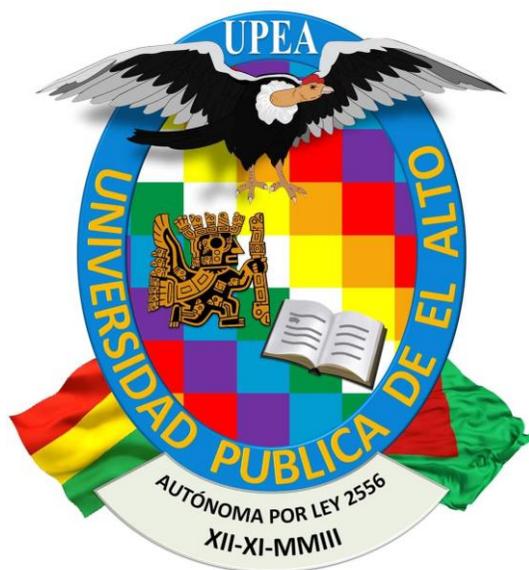


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“APLICACIÓN WEB PARA LA GESTIÓN DE PROCESOS EDUCATIVOS”

CASO: (UNIDAD EDUCATIVA “CALAMA”)

Para optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: GESTIÓN Y PRODUCCIÓN

Postulante: David Gonzalo Cusi Quispe

Tutor Metodológico: Ing. Maricel Yarari Mamani

Tutor Revisor: Ing. Sergio Ramiro Rojas Saire

Tutor Especialista: M.Sc. Reynaldo Javier Zeballos Daza

EL ALTO – BOLIVIA

2020

DEDICATORIA

A mis padres Benigno y Eugenia y a mi hermano Álvaro quienes me brindaron todo su amor, cariño, y comprensión y quienes me inculcaron por el buen camino con sus consejos y recomendaciones que me dieron y por el apoyo constante que me brindaron en el desarrollo del proyecto.

A mis amigos por permitirme aprender más de la vida a su lado

AGRADECIMIENTOS

A mis padres Benigno y Eugenia, que siempre estuvieron apoyándome en todo momento, gracias por todo su esfuerzo y sacrificio para ayudarme a salir adelante, sin Uds. no fuese posible todo este logro, gracias por tanto amor.

A mi tutor M. Sc. Reynaldo Javier Zeballos Daza, quien me brindó todo su conocimiento y tiempo, para poder realizar este proyecto, gracias por su paciencia, sus enseñanzas y su comprensión.

Al Ing. Sergio Ramiro Rojas Saire y a la Ing. Maricel Yarari Mamani, por su ayuda y comprensión, quienes me ayudaron en todo el proceso de este proyecto con su conocimiento y sus consejos.

Al Lic. Wilfredo Villegas Alvares Director de la Unidad Educativa Calama, quien me permitió realizar este proyecto en la, gracias por sus ánimos, sus consejos y su ayuda.

A todos mis amigos, compañeros que me tuvieron paciencia, me animaron y ayudaron en esta etapa de mi vida.

A todos ellos, estaré eternamente agradecido, muchas gracias.

INDICE

CAPITULO I.....	1
1. MARCO PRELIMINAR.....	1
1.1 INTRODUCCIÓN	1
1.2 ANTECEDENTES.....	2
1.2.1 Antecedentes de la Institución	2
1.2.2 Antecedentes de trabajos afines	3
1.3 PLANTEAMIENTO DEL PROBLEMA	4
1.3.1 Problema Principal.....	4
1.3.2 Problemas Secundarios	4
1.3.3 Formulación del Problema	5
1.4 OBJETIVOS.....	5
1.4.1 Objetivo General.....	5
1.4.2 Objetivos Específicos.....	5
1.5 JUSTIFICACIÓN.....	6
1.5.1 Justificación Económica.....	6
1.5.2 Justificación Técnica.....	6
1.5.3 Justificación Social	6
1.6 METODOLOGÍA	6
1.7 TÉCNICAS DE INVESTIGACIÓN	8
1.7.1 Observación	8
1.7.2 Entrevistas.....	8
1.7.3 Encuestas.....	8
1.8 HERRAMIENTAS	8

1.8.1 Lenguaje de programación JavaScript.....	9
1.8.2 Framework React	9
1.8.3 Base de Datos Cloud Firestore	9
1.8.4 Maquetación	10
1.8.5 Git y Github.....	10
1.9 LÍMITES Y ALCANCES	10
1.9.1 Límites.....	10
1.9.2 Alcances	10
1.9.2.1 Módulos.....	10
1.9.2.2 Funcionalidades	11
1.10 APORTES.....	11
CAPÍTULO II.....	12
2. MARCO TEÓRICO	12
2.1 PLATAFORMA EDUCATIVA	12
2.2 PLANIFICACIÓN EDUCATIVA	12
2.3 EVALUACIÓN DE VIRTUAL.....	12
2.3.1 Registro pedagógico	12
2.4. EVALUACIÓN DE LAS DIMENSIONES EN ESTUDIANTES.....	13
2.4.1 Saber.....	13
2.4.2 Hacer	13
2.4.3 Ser y Decidir	13
2.5 MÉTODO KAIZEN	13
2.6 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	14
2.6.1 Metodologías Ágiles.....	14

2.6.1.1 Principios del Manifiesto Ágil	14
2.7 METODOLOGÍA KANBAN.....	15
2.7.1 Roles Kanban	16
2.7.2 Principios de la Metodología Kanban.....	16
2.7.2.1 Paso 1: Capturar la rutina de trabajo	16
2.7.2.2 Paso 2: Publicar el <i>workflow</i> (Flujo de trabajo) en un tablero.....	17
2.7.2.3 Paso 3: Establecer límites al trabajo en curso (WIP)	18
2.7.2.4 Paso 4: Definir el estado de Hecho.....	19
2.7.2.5 Paso 5: Ejecuta tu standup diario	19
2.7.3 Métricas en Kanban	19
2.8 UML.....	20
2.8.1 Diagramas de clases	20
2.8.2 Diagramas de casos de uso.....	21
2.8.3 Diagramas de interacción	22
2.8.2 Diagramas de estado.....	23
2.9 PATRONES DE DISEÑO.....	23
2.9.1 Modelo Vista Controlador	24
2.10 TECNOLOGÍAS WEB	25
2.10.1 Framework React	25
2.10.2 JavaScript.....	26
2.10.3 Backend como un servicio (BaaS Backend as a Service)	27
2.10.3.1 Cloud Firestore	27
2.10.3 Bases de Datos NoSQL.....	28
2.11 LEARNING MANAGMENT SYSTEM (LMS)	29

2.12 METODOLOGÍA WEBQEM	29
2.12.1 Definición de las metas de evaluación y selección del perfil de usuario ..	33
2.12.2 Definición de criterios elementales y procedimientos de medición	34
2.12.3 Definición de agregación e implementación de la evaluación global	34
2.12.4 Análisis de resultados y recomendaciones	34
2.13 COCOMO	34
2.14 CASOS DE PRUEBA.....	35
2.15 PRUEBAS UNITARIAS	36
2.16 SEGURIDAD.....	36
2.16.1 Principios de Seguridad	36
2.16.2 Seguridad Lógica	36
2.16.3 Seguridad Física	37
CAPITULO III.....	38
3. DISEÑO METODOLÓGICO.....	38
3.1 INTRODUCCIÓN	38
3.2 ANALISIS DE LA INSTITUCIÓN	38
3.3 COBINACIÓN DE KANBAN CON UML	39
3.4. KANBAN ROLES	39
3.4 MÓDULO DE ADMINISTRACIÓN DE USUARIOS	40
3.4.1 Fase de Pedidos	40
3.4.1.1 Tablero Kanban	40
3.4.2 Fase de Especificación	41
3.4.2.1 Casos de uso: Administrar Usuarios	41
3.4.2.2 Diagrama de Clases: Administrar Usuarios	44

3.4.3 Fase de Implementación.....	45
3.4.3.1 Diagrama de estado: Administrar Usuarios.....	45
3.4.3.2 Tarea: Crear cuenta	46
3.4.3.3 Tarea: Inicio de sesión.....	47
3.4.3.4 Tarea: Editar Cuenta	48
3.4.3.5 Tarea: Ver, Crear, Editar y Eliminar Cuentas	48
3.4.4 Fase de Validación	49
3.4.4.1 Pruebas unitarias.....	50
3.5 MÓDULO DE SERVICIOS	51
3.5.1 Fase de Pedidos	51
3.5.2 Fase de Especificación	52
3.5.2.1 Casos de Uso: Operaciones del Administrativo	52
3.5.2.2 Casos de Uso: Operaciones del Docente	53
3.5.2.3 Casos de Uso: Operaciones del Estudiante.....	58
3.5.2.4 Diagrama de Clases: Módulo de Servicios	62
3.5.2.5 Tablero Kanban	63
3.5.3 Fase de Implementación.....	65
3.5.3.1 Diagrama de estado: Operaciones del Administrativo	65
3.5.3.2 Diagrama de estado: Operaciones del Docente	65
3.5.3.2 Diagrama de estado: Operaciones del Estudiante	66
3.5.4 Fase de Validación	67
3.5.4.1 Pruebas unitarias.....	68
3.6 MÓDULO DE REPORTES.....	69
3.6.1 Fase de Pedidos	69

3.6.2 Fase de Especificación	69
3.6.2.1 Casos de Uso: Reportes para Usuarios.....	69
3.6.2.2 Tablero Kanban	71
3.6.3 Fase de Implementación.....	72
3.6.3.1 Diagrama de estado: Generar Reportes	72
3.6.4 Fase de Validación	72
3.6.4.1 Pruebas unitarias.....	73
CAPITULO IV	74
4. CALIDAD Y SEGURIDAD.....	74
4.1 CALIDAD ISO 9126	74
4.1.1 Funcionalidad	74
4.1.2 Usabilidad.....	75
4.1.3 Mantenibilidad.....	76
4.1.4 Confiabilidad	76
4.1.5 Portabilidad.....	77
4.1.4 Calidad Global	78
4.2 SEGURIDAD	79
4.2.1 Seguridad Física.....	79
4.2.2 Seguridad Lógica.....	79
4.2.2.1 Integridad	79
4.2.2.2 Confidencialidad	79
4.2.2.3 Disponibilidad	79
CAPITULO V	81
5. ANALISIS DE COSTOS.....	81

5.1 COSTO DEL SOFTWARE	81
5.1.1 Punto de Función.....	81
5.1.1.1 Número de entradas de usuario	81
5.1.1.2 Número de salidas de usuario	82
5.1.1.3 Número de peticiones de usuario	82
5.1.1.4 Numero de interfaces externas.....	84
5.1.2 Conversión de PF a LDC (líneas de código)	86
5.1.3 Estimaciones de esfuerzo nominal.....	87
5.1.4 Estimación de duración del proyecto	91
5.1.5 Costo de desarrollo.....	92
5.1.6 Costo de implementación	93
5.1.7 Costo de elaboración	93
5.1.8 Costo total de elaboración del proyecto	93
CAPITULO VI	95
6. CONCLUSIONES Y RECOMENDACIONES	95
6.1 CONCLUSIONES	95
6.2 RECOMENDACIONES	95
BIBLIOGRAFIA.....	97
ANEXOS.....	100

INDICE DE TABLAS

Tabla 1 Fases de Trabajo, WIP y responsables	39
Tabla 2: Especificación del caso de uso Crear Cuenta	41
Tabla 3: Especificación de casos de uso Editar Cuenta y Eliminar Cuenta	42
Tabla 4: Especificación del caso de Uso Cambiar Roles de Estudiantes y Docentes	43
Tabla 5: Especificación del caso de uso Ver, Crear, Editar y Eliminar Cuentas	43
Tabla 6: Validación del Módulo Administración de Usuarios	49
Tabla 7: Pruebas del Módulo Administración de Usuarios	50
Tabla 8: Especificación del caso de uso Cambiar de Curso a estudiante.....	52
Tabla 9: Especificación del caso de uso Ver Cursos Propios	53
Tabla 10: Especificación del caso de uso Crear Curso	54
Tabla 11: Especificación del caso de uso Editar Curso	55
Tabla 12: Especificación del caso de uso Calificar.....	56
Tabla 13: Especificación de Caso de Uso Crear y Publicar Evaluaciones.....	57
Tabla 14: Especificación del caso de uso Ver lista de cursos asignados	58
Tabla 15: Especificación del caso de uso Crear una pregunta	59
Tabla 16: Especificación del caso de uso Ver Calificación	60
Tabla 17: Ver y Realizar evaluación.....	61
Tabla 18: Validación de Módulo de Servicios.....	67
Tabla 19: Pruebas Unitarias Módulo de Servicios	68
Tabla 20: Especificación del caso de uso Generar Lista de Estudiantes.....	70
Tabla 21: Especificación del caso de uso Registro de Evaluación Trimestral	70
Tabla 22: Validación de Módulo de Reportes.....	72

Tabla 23: Pruebas Unitarias Módulo de Reportes	73
Tabla 24: Prueba de Funcionalidad del Sistema	74
Tabla 25: Encuesta de usabilidad del sistema	75
Tabla 26: Mantenibilidad del sistema	76
Tabla 27: Calidad Global.....	78
Tabla 28: Entradas de usuario	81
Tabla 29: Salidas de usuario.....	82
Tabla 30: Peticiones de usuario	82
Tabla 31: Número de archivos	83
Tabla 32: Número de interfaces externas	84
Tabla 33: Factor de ajuste de complejidad.....	84
Tabla 34: Tabla de conversión factor LDC	86
Tabla 35: Factores de escala W_j	88
Tabla 36: Multiplicadores de esfuerzo.....	89
Tabla 37: Multiplicadores de esfuerzo de plataforma	89
Tabla 38: Multiplicadores de esfuerzo de Personal	89
Tabla 39: Multiplicadores de esfuerzo del proyecto	90
Tabla 40: Multiplicadores para el costo de proyecto	90
Tabla 41: Costo de Elaboración	93
Tabla 42: Costo total del proyecto.....	93

INDICE DE FIGURAS

Figura 1: Ejemplo de Rutina de trabajo para desarrollo de software	16
Figura 2: Tablero Kanban	18
Figura 3: Tablero Kanban con tareas colocadas	18
Figura 4: Limites del WIP.....	19
Figura 5: Métricas Kanban	20
Figura 6: Diagrama de clase	21
Figura 7: Diagrama de casos de uso	22
Figura 8: Diagrama de interacción	23
Figura 9: Diagrama de estado.....	23
Figura 10: Tablero Kanban columna de pedidos	40
Figura 11: Casos de uso: Administrar Roles de Usuario	41
Figura 12: Diagrama de Clases: UsuariosFirebase y users.....	44
Figura 13: Tablero Kanban columna de especificación	45
Figura 14: Diagrama de estado Administración de Usuarios.....	46
Figura 15: Interfaz Crear Cuenta.....	47
Figura 16: Interfaz Inicio de Sesión.....	47
Figura 17: Interfaz Editar Cuenta	48
Figura 18: Interfaz Editar y Eliminar cuentas.....	49
Figura 19: Resultado de pruebas unitarias módulo de administración de usuarios	50
Figura 20: Casos de Uso: Operaciones del Administrativo.....	52
Figura 21: Casos de Uso: Operaciones del Docente.....	53
Figura 22: Casos de Uso: Operaciones del Estudiante	58

Figura 23: Casos de Uso: Operaciones del Docente.....	63
Figura 24: Tablero Kanban Tareas para el módulo de servicios	64
Figura 25: Diagrama de estado: Operaciones de Administrativo.....	65
Figura 26: Diagrama de estado: Operaciones de Docente.....	66
Figura 27: Diagrama de estado: Operaciones de Estudiante	67
Figura 28: Resultado de pruebas unitarias módulo de servicios	68
Figura 29: Caso de Uso: Reportes para el Personal Administrativo	69
Figura 30: Tablero: Módulo de Reportes.....	71
Figura 31: Diagrama de estado: Generar Reportes.....	72
Figura 32: Resultado de pruebas unitarias módulo de reportes	73

CAPITULO I

1. MARCO PRELIMINAR

1.1 INTRODUCCIÓN

Las actividades escolares fueron paralizadas desde marzo en Bolivia a causa del coronavirus; pero algunas escuelas y universidades continuaron con el avance de forma virtual. Debido a la cuarentena, varios estudiantes han optado por las clases virtuales como alternativa a continuar con la gestión escolar, pero la realidad es que la mayoría de las Unidades Educativas en Bolivia no están preparadas para pasar clases virtuales ya que no se cuenta con aplicaciones web en la nube que gestionen los procesos que se llevan a cabo en el desarrollo de contenidos.

Los retos para la unidad educativa son que el aprendizaje continúe para los estudiantes y que la diferencia entre las clases presenciales y virtuales no dificulte el proceso de aprendizaje y enseñanza. También es complicado pasar de un sistema presencial a un sistema de aulas virtuales en cuanto a la planificación y organización, ya que no se cuenta con sistemas capaces de organizar y centralizar las clases virtuales para tener una transición más cómoda para los estudiantes, padres de familia y profesores.

Los administrativos y docentes en esta nueva realidad tienen como tareas las de capacitarse en el uso de herramientas digitales para reorganizar sus contenidos en las materias escolares, planificar sus clases y llevar un control y seguimiento a sus estudiantes. Eso implica que un profesor tiene muchas nuevas tareas que cumplir y tratar que los estudiantes no se perjudiquen y que las clases virtuales sean comprensibles.

Los estudiantes tienen que enfrentar las tareas de organizar su horarios y clases virtuales de cada una de las materias en la escuela o colegio teniendo que contar con acceso a un dispositivo móvil como el celular o de escritorio como el computador personal y tener que pagar el internet. Esto significa una tarea muy complicada para los estudiantes que no cuentan con experiencia previa con clases virtuales que no son dictadas únicamente por un solo profesor, sino que tienen que pasar clases

virtuales con diferentes profesores en todas las asignaturas y además con diferentes plataformas virtuales.

El problema de las escuelas al no contar con una única plataforma implica que tienen desorganizado y descentralizado en distintas plataformas los contenidos y las clases virtuales, lo que no permite el correcto avance de los contenidos curriculares en las diferentes materias. También implica un trabajo extra para los profesores que tienen que incluir a sus estudiantes en sus propias plataformas de manera manual y poco eficiente.

El contar con una plataforma educativa permite a las Unidades Educativas organizar de mejor manera las aulas virtuales de todos los estudiantes y profesores. El tener centralizado una agenda o cronograma virtual para desarrollar el curso, permite tener mejores logros de aprendizaje o competencias, tener un banco de recursos y documentación relevante, implementar de mejor manera estrategias de evaluación y tener información sobre el rendimiento académico de los estudiantes.

Los resultados obtenidos con el presente proyecto de grado mostrarán los beneficios en el tiempo de trabajo para la gestión de procesos recurrentes en la Unidad Educativa “Calama” como ser: Organización de contenidos, inscripción de estudiantes, publicación de horarios, comunicados y la centralización de las actividades acordes a la propia organización de la institución.

1.2 ANTECEDENTES

Según el decreto supremo 4260 promulgado por el Ministerio de Educación, Culturas y Deportes se norma la complementariedad de las modalidades de atención presencial, a distancia, virtual y semipresencial en los subsistemas de Educación. (Ministerio de Educación, Deportes y Culturas, 2020, pp. 4-6)

1.2.1 Antecedentes de la Institución

La unidad educativa “Calama” se encuentra ubicada en la ciudad de El Alto en la zona Pacajes Caluyo frente a la plaza Bolivia. Esta Unidad Educativa tiene el nivel de Secundaria Comunitaria Productiva en el horario de la tarde.

La unidad Educativa cuenta con todos los cursos de nivel secundario y cinco paralelos en cada curso de secundaria.

Actualmente la Unidad Educativa cuenta con más de 700 estudiantes y 48 docentes. El director de la Unidad Educativa es el Sr. Lic. Wilfredo Villegas Álvarez.

1.2.2 Antecedentes de trabajos afines

“Learning Management System (LMS) Moodle”. Es en la actualidad el Sistema de gestión de aprendizaje más usado en todo el mundo. Este sistema es de código abierto. Brinda soluciones para escuelas, educación superior y para el lugar de trabajo. Moodle está construido principalmente en el lenguaje PHP y AJAX. (Moodle Pty Ltd, 6 de abril 2020).

(Rentería, 2015). “Implementación de una plataforma virtual como estrategia metodológica que permita mejorar el rendimiento académico en el área de matemática (..)”. La propuesta es la utilización del software libre Edmodo para adaptarlo e implementarlo en la Unidad Educativa Chigorodo para ver una mejora en el aprendizaje de matemáticas.

(Dávila, 2013). “Análisis, diseño e implementación de un sistema de aula virtual para captación de personal en la empresa Undermedia S.A.”. Se implementa un sistema prototipo con un servidor con toda la documentación y manuales de usuario. La metodología usada en el proyecto es SCRUM. Las herramientas utilizadas son LMS Moodle y una base de datos MYSQL.

“Plataforma colaborativa de cursos masivos en línea”. Universidad Mayor de San Andrés. La Paz, Bolivia. Propone un modelo de plataforma colaborativa de cursos en línea para que estudiantes tengan acceso y organización de contenidos, a mecanismos para el aprendizaje, comunicación formativa y colectiva. (Apaza, 2016).

1.3 PLANTEAMIENTO DEL PROBLEMA

1.3.1 Problema Principal

No existe un sistema de gestión que contemple los procesos educativos como actividades curriculares, clases virtuales, evaluación de actividades acorde a los criterios y organización de la Unidad Educativa “Calama”. Esto implica un trabajo extra y poco organizado para los docentes en cada una de las materias o áreas educativas.

1.3.2 Problemas Secundarios

Se mencionan los problemas más destacados:

- Inexistencia de un sistema de gestión sobre los procesos educativos que sea accesible para administrativos, docentes y estudiantes.
- El proceso de recopilar los datos necesarios para las plataformas educativas es realizado por cada docente en los distintos cursos y no existe un sistema que se encargue de recopilar toda la información y repartir esta información a cada docente.
- El proceso de inscripción de estudiantes a las aulas virtuales, son repetidos para cada área educativa, esto implica un proceso lento en la inserción de los estudiantes.
- Al tener múltiples plataformas educativas en las distintas materias hace confuso y complicado la organización para los estudiantes y docentes.
- Existe poca retroalimentación para los estudiantes sobre los contenidos curriculares avanzados.
- Los documentos digitales utilizados por los docentes y estudiantes están almacenados en distintas plataformas lo que dificulta su organización. Por el volumen de personas y de documentos que se manejan el acceso a la información es limitado.
- Para realizar la evaluación tomando en cuenta las tres dimensiones de evaluación: Saber, Hacer y Ser/Decidir el docente tiene que hacer una previa configuración en las distintas plataformas educativas.

- La configuración sobre la evaluación Saber, Hacer y Ser/Decidir, es de forma manual en las distintas plataformas educativas , puesto que no es una plataforma especial o adecuada a la Unidad Educativa y a los registros pedagógicos de cada docente.
- El docente debe de adicionar las actividades que se realizan de manera virtual al registro de evaluación lo que significa un trabajo extra para el docente.
- Los estudiantes tienen la preferencia de participar en las actividades virtuales con el envío de mensajes de texto y muchas veces estos mensajes no son tomados en cuenta para el registro de evaluación.

1.3.3 Formulación del Problema

¿De qué manera se puede organizar y gestionar las actividades curriculares, clases virtuales y evaluaciones en la Unidad Educativa “Calama”?

1.4 OBJETIVOS

1.4.1 Objetivo General

Implementar una Aplicación Web para la gestión de procesos educativos, que permita almacenar, organizar y gestionar las actividades curriculares de manera eficiente, acorde a los criterios y organización de la Unidad Educativa “Calama”.

1.4.2 Objetivos Específicos

- Analizar el flujo de información y requerimientos en estudiantes, docentes y administrativos.
- Diseñar y modelar la estructura de la aplicación web y la base de datos.
- Generar un sistema de inscripción que optimice los procesos de carácter académico y administrativo para los estudiantes.
- Organizar y sistematizar la visualización de contenidos curriculares para tener una administración productiva sobre la información necesaria para cada estudiante y docente.
- Simplificar el proceso de evaluación y generar reportes de calificaciones acordes a los criterios de evaluación.

1.5 JUSTIFICACIÓN

1.5.1 Justificación Económica

Se utilizará en un principio el hosting gratuito de Firebase Hosting y según al requerimiento de la Unidad Educativa se buscará un plan de uso de hosting por cantidad de tráfico en el sistema.

Al ser un sistema web alojado en un servidor externo, no se requerirá la compra de equipos de computación nuevos ya que estos sistemas web pueden ejecutarse tanto en dispositivos móviles y de escritorio mediante un navegador web como Chrome, Firefox y otros.

1.5.2 Justificación Técnica

La base de datos Firestore de Firebase cuenta con un plan de uso gratuito inicial que puede ser esacalado y si el sistema requiere de un mayor espacio de almacenamiento de datos, esto permite al sistema garantizar la disponibilidad al permitir soporte para mayor cantidad de datos.

El sistema web estará alojado en el hosting gratuito de *firebase* que permitirá al sistema estar a disposición de la Unidad Educativa las 24 horas del día. Esto permitirá mejorar y agilizar procesos de inserción y organización de clases virtuales para docentes y estudiantes.

El dominio de la aplicación web es otorgado por el mismo servicio de base de datos y cuenta con un certificado SSL que garantiza al usuario que los datos recibidos y enviados están cifrados tanto del servidor al cliente o del cliente al servidor.

1.5.3 Justificación Social

Con la implementación del sistema web se pretende dar un servicio de información de calidad y confiable que beneficie a toda la comunidad educativa tanto al director, plantel docente, estudiantes y padres de familia.

1.6 METODOLOGÍA

Para el desarrollo del Sistema se utilizará la metodología ágil Kanban que permite visualizar el flujo de trabajo, dividiendo el trabajo en bloques, cada tarea a realizar

es descrito en una tarjeta para luego ponerlo en el cuadro Kanban. Kanban permite utilizar columnas con nombre para ilustrar dónde está cada tarea en el flujo de trabajo. Limita el trabajo en curso, donde se asigna límites concretos a cuantas tareas pueden estar en progreso en cada estado del flujo de trabajo. (Brechner, 2015, capítulo 1)

Las etapas del desarrollo Kanban son:

- Capturar la rutina de alto nivel del equipo.
- Establecer etapas en Tablero Kanban.
- Establecer límites.
- Definir el estado de hecho.
- Ejecutar standup diario.

También se utiliza el método Kaizen que está muy relacionado a la forma de trabajo de Kanban y en la mejora continua de proyectos.

En las aplicaciones técnicas y orientadas a la producción de Kaizen, los pequeños pasos se consideran una forma de producir un cambio acumulativo. Un pequeño cambio tras otro pequeño cambio seguido de otro pequeño cambio, y así sucesivamente, eventualmente lo depositarán en el umbral de su objetivo. Este proceso se resume en el dicho del antiguo filósofo chino Lao Tse: "Un viaje de mil millas comienza con el primer paso". (Maurer, 2012).

Para el modelado y análisis del sistema se utiliza UWE que sirve para modelar aplicaciones web, presta una especial atención a la sistematización y personalización. Provee de perfiles UML, meta modelos, un proceso de desarrollo dirigido para modelos, y herramientas de soporte para el diseño sistemático de aplicaciones web. Utiliza notación basada en UML para aplicaciones web en general y para aplicaciones adaptativas en particular. La metodología consta de seis modelos:

- Modelo de casos de uso para capturar los requisitos del sistema.
- Modelo conceptual para el contenido (modelo del dominio).

- Modelo de usuario: modelos de navegación que incluye modelos estáticos y dinámicos.
- Modelo de estructura de presentación, modelo de flujo de presentación.
- Modelo abstracto de interfaz de usuario y modelo de ciclo de vida del objeto.
- Modelo de adaptación.

1.7 TÉCNICAS DE INVESTIGACIÓN

Las técnicas de investigación que se utilizarán para plantear el proyecto serán: la observación para el análisis, entrevista y encuestas para obtener información de requerimientos.

1.7.1 Observación

Fundamental en todo principio científico, la observación consiste en simplemente confrontar el fenómeno que se desea comprender y describirlo, tomar nota de sus peculiaridades, de su entorno, en fin, detallarlo. Suele ser el primer paso básico de todo tipo de saber. (Raffino, 2020).

1.7.2 Entrevistas

Consiste en hacerle preguntas directamente al sujeto o los sujetos de estudio, generalmente en un lugar aislado, para así obtener una aproximación a lo que piensa, siente o ha vivido, que luego podrá ser procesada estadísticamente o mediante otros métodos, para obtener una verdad. (Raffino, 2020).

1.7.3 Encuestas

La encuesta es un instrumento para recoger información cualitativa y/o cuantitativa de una población estadística. Para ello, se elabora un cuestionario, cuyos datos obtenidos será procesados con métodos estadísticos. (Raffino, 2020).

1.8 HERRAMIENTAS

Para la codificación de este sistema web se utilizará las siguientes herramientas tecnológicas:

1.8.1 Lenguaje de programación JavaScript

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB and Adobe Acrobat. JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo programación funcional). (Mozilla, 2020).

Se utilizará el lenguaje de programación Javascript para desarrollar el código y librerías necesarias para la conexión de la página web con el servidor.

1.8.2 Framework React

El proyecto utilizará la forma de creación basado en el framework react js que permitirá un mejor orden y con menos código de javascript.

React es una biblioteca Javascript para crear interfaces de usuario. React ha sido diseñado desde su inicio para ser adoptado gradualmente, así puedes usar tan poco o mucho de React como necesites. (Facebook Inc, 2020).

1.8.3 Base de Datos Cloud Firestore

Se utilizará la base de datos no relacional Cloud Firestore para tener sincronizado los datos del sistema web.

Cloud Firestore es una base de datos flexible y escalable para la programación en servidores, dispositivos móviles y la Web desde Firebase y Google Cloud Platform. Al igual que Firebase Realtime Database, mantiene tus datos sincronizados entre apps cliente a través de agentes de escucha en tiempo real y ofrece asistencia sin conexión para dispositivos móviles y la Web, por lo que puedes compilar apps con capacidad de respuesta que funcionan sin importar la latencia de la red ni la conectividad a Internet. Cloud Firestore también ofrece una integración sin interrupciones con otros productos de Firebase y Google Cloud Platform, incluido Cloud Functions. (Firebase, 2020).

1.8.4 Maquetación

En el inicio del proyecto se desarrollará la maquetación del proyecto en el software Adobe XD para luego pasarlo a HTML5 y CSS3.

1.8.5 Git y Github

GitHub facilita la colaboración con git. Es una plataforma que puede mantener repositorios de código en almacenamiento basado en la nube para que varios desarrolladores puedan trabajar en un solo proyecto y ver las ediciones de cada uno en tiempo real.

1.9 LÍMITES Y ALCANCES

1.9.1 Límites

El sistema web tiene los siguientes límites

- Solo usuarios habilitados por el administrador del sistema tendrán acceso al sistema.
- El sistema se encarga solo de la organización y control de clases virtuales.
- El sistema no podrá funcionar en lugares donde no se cuente con conexión a internet.
- El sistema sólo podrá ser controlado a través de celulares, ordenadores que tengan una conexión a internet.
- El sistema web solo mostrará información personal a los propietarios de dicha información que tengan una cuenta habilitada.

1.9.2 Alcances

Basado en las principales necesidades de la empresa, el sistema cuenta con los siguientes módulos y funcionalidades.

1.9.2.1 Módulos

- Módulo de administración de usuarios.
- Módulo de administración de información
- Módulo de reporte de datos

1.9.2.2 Funcionalidades

- Funcionalidad responsiva, que permite al sistema ser visualizado de manera correcta en dispositivos móviles.
- Funcionalidad de seguimiento de conexiones de clientes que permita ver el ultimo uso por usuario.
- Funcionalidad de registro de profesores, cursos, materias.
- Funcionalidad de organización automática por niveles de aulas virtuales.
- Funcionalidad de publicación de notas trimestrales por estudiantes.

1.10 APORTES

El sistema propuesto, es de mucha utilidad y es necesario porque brinda información, simplifica y mejora las actividades de las clases virtuales al tener un mayor control y organización para los estudiantes y profesores.

Así mismo, el sistema coadyuva a mejorar los procesos y actividades para todos los usuarios de la institución educativa.

El sistema combina las tecnologías de bases de datos *NoSql* y el *framework* de desarrollo *React* obteniendo de esta manera un alto rendimiento y respuesta a las peticiones de todos los usuarios.

CAPÍTULO II

2. MARCO TEÓRICO

A continuación, se describe la teoría sobre las metodologías, técnicas y herramientas que se utiliza para el desarrollo del sistema, sin embargo, solo contiene los conceptos más relevantes para que pueda aclarar cualquier interrogante.

2.1 PLATAFORMA EDUCATIVA

Es un espacio virtual de aprendizaje conectado a internet o intranet organizado como un sistema de información, planificación y comunicación para la ejecución de diversas herramientas y aplicaciones que permite a un maestro contar con un aula virtual donde coloca los materiales de formación, enlaza a otros recursos educativos, recibe tareas de los estudiantes, evalúa los trabajos, promueve debates y cuenta con estadísticas de evaluación, reporte de asistencias y rendimiento escolar. (Ministerio de Educación, Deportes y Culturas, 2020, pp. 4-6).

2.2 PLANIFICACIÓN EDUCATIVA

El maestro puede utilizar recursos educativos como la documentación relevante, material de apoyo, bibliografía complementaria, webgrafía, taller multimedia, cartelera fílmica, actividades prácticas, ejercicios y enlaces a espacios de aprendizaje para el desarrollo de las estrategias metodológicas. (Ministerio de Educación, Deportes y Culturas, 2020, p. 6).

2.3 EVALUACIÓN DE VIRTUAL

Cada contenido temático desarrollado por los maestros debe concluir con una actividad evaluativa que permita valorar los aprendizajes de los estudiantes (..). (Ministerio de Educación, Deportes y Culturas, 2020, p. 8).

2.3.1 Registro pedagógico

El Registro Pedagógico es un documento escrito, con el formato de una narrativa personal, que va más allá de contar la tarea de investigación de los estudiantes, se

centra en la descripción de las acciones de enseñanza (Ministerio de Educación, Deportes y Culturas, 2020, p. 10).

2.4. EVALUACIÓN DE LAS DIMENSIONES EN ESTUDIANTES

2.4.1 Saber

Hace referencia al aspecto cognitivo en el sentido de conocer conceptualizar, analizar, comprender los conocimientos de la realidad vinculados a los conocimientos científicos. (Ministerio de Educación, Deportes y Culturas, 2020, p. 10).

2.4.2 Hacer

Manifiesta énfasis en el proceso y hace referencia a las capacidades, habilidades y destrezas, que permite el dominio de la técnica, habilidad o estrategia del conocimiento adquirido, como resolución de problemas, proyectos, ejecución de una tarea práctica o intelectual, procedimientos a seguir. (Ministerio de Educación, Deportes y Culturas, 2020, p. 10).

2.4.3 Ser y Decidir

Estos constituyen principios, valores, normas, creencias y actitudes, que llevan al equilibrio personal en la convivencia social, por medio de la toma de decisiones. (Ministerio de Educación, Deportes y Culturas, 2020, p. 10).

2.5 MÉTODO KAIZEN

En Kaizen se trabaja mediante la realización de pequeños pasos acumulativos que nos llevarán a nuestro objetivo.

Un enfoque Kaizen le pide que dé pequeños pasos hacia su objetivo. Estos pasos son tan pequeños que pueden parecer inútiles, pero por eso funcionan. (Maurer, 2012, Capítulo uno).

Al aplicar el método Kaizen se busca lograr:

- Elevar la moral.
- Contener costos.

- Mejorar la calidad.
- Desarrollar nuevos productos y servicios.
- Reducir los gastos. (Maurer, 2012, Capítulo uno)

2.6 METODOLOGÍAS DE DESARROLLO DE SOFTWARE

El éxito en el proyecto, dependerá en gran parte de escoger una metodología que potencie la calidad del producto final. (Cabrera, Figueroa, y Solís, 2008).

2.6.1 Metodologías Ágiles

Existen variadas metodologías ágiles que están basadas en una mentalidad “AGILE”. Stellman y Greene (2014) afirma: “Agile es un conjunto de métodos y metodologías que ayudan a su equipo a pensar de manera más efectiva, trabajar de manera más eficiente y tomar mejores decisiones”.

Cada una de estas metodologías cuenta con prácticas que están simplificadas y optimizadas para una mejor adopción en el desarrollo de un proyecto.

Se cuenta con un Manifiesto por el desarrollo Ágil de software redactado por expertos en programación para hacer un cambio radical en la forma de desarrollar “software”.

2.6.1.1 Principios del Manifiesto Ágil

Medinilla, Giné y Gómez (2001) nos muestran los doce principios del manifiesto ágil en una traducción al español:

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
- Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.

- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El software funcionando es la medida principal de progreso.
- Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

2.7 METODOLOGÍA KANBAN

La Metodología Kanban tiene su origen en los procesos de producción “Justo a Tiempo” de Toyota, en donde se utilizaban tarjetas para marcar la necesidad de material de producción, donde los trabajadores producen su parte siempre y cuando les llegue una tarjeta que les informe que pueden producir. De esta manera están limitados y prohibidos de producir su parte del trabajo si es que no les llega su tarjeta.

Kanban forma parte de las metodologías ágiles, cuyo objetivo es gestionar de manera general cómo completar las tareas en un proyecto.

Derivado de la combinación de dos palabras japonesas, kan, que quiere decir visual, y ban, que quiere decir tarjeta, y esto forma la palabra Kanban que denomina una metodología que se basa en tarjetas visuales para gestionar el trabajo y dedicación del equipo de proyecto (Bermejo, 2015).

Las ventajas de la metodología Kanban es que es muy fácil de utilizar, actualizar en un proyecto. Esto sucede porque es muy visual, que permite ver el estado del proyecto y también pautar el desarrollo del trabajo de manera efectiva.

Kanban comparte con otras metodologías ágiles la idea de un Product Backlog (Lista de trabajos pendientes) que tiene una serie de ítems priorizados. Un de las principales diferencias es que no existen las iteraciones. Kanban se centre en controlar el *Work In Progress* (trabajo en curso).

2.7.1 Roles Kanban

En la metodología Kanban asignar roles no es de carácter obligatorio, así como también las reuniones no son normadas. Pero se han ido creando y adoptando algunos roles observados en organizaciones que han utilizado el método (Palacios, 2016)

2.7.2 Principios de la Metodología Kanban

La metodología Kanban está basado en principios que lo caracterizan y resaltan de las otras metodologías ágiles. A continuación, se tratará de explicar los pasos para aplicar la metodología kanban.

2.7.2.1 Paso 1: Capturar la rutina de trabajo

Se realizan muchos tipos de trabajo:

- Analizar el producto con todas las partes interesadas.
- Encontrar, evaluar y corregir errores y problemas operativos.
- Producir mejoras a productos e infraestructura.
- Redacción de propuestas o documentos de diseño importantes.

En la Figura 1 muestra una rutina de trabajo utilizada mayormente y adoptada por muchos equipos.

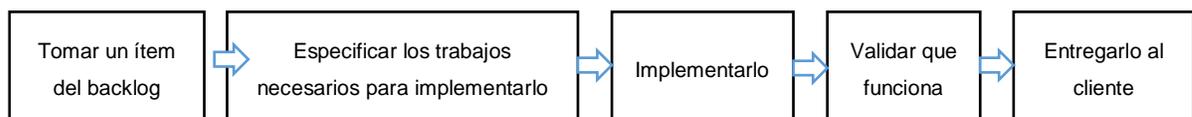


Figura 1: Ejemplo de Rutina de trabajo para desarrollo de software
Fuente: Brechner, E. Agile Project Management with Kanban. 2015

Una vez que se tiene la rutina de trabajo de las personas lo que se debe hacer es especificar en pequeños pasos y de igual tamaño un elemento de trabajo del backlog.

Los equipos que utilizan métodos muy formales (quizás para el cumplimiento) o que realizan un diseño en profundidad pueden utilizar varios pasos para especificar un elemento de trabajo. Los equipos que tienen procedimientos o estándares complejos pueden utilizar varios pasos para implementar o validar un elemento de trabajo. (Brechtner, 2015, capítulo 1)

2.7.2.2 Paso 2: Publicar el *workflow* (Flujo de trabajo) en un tablero

En este paso se representará de forma física y visual el flujo de trabajo que permitirá ver la realidad del proyecto a cada momento.

El objetivo de esto es:

- Entender mejor los procesos que se realizan.
- Conocer los problemas que puedan surgir y solucionarlas.
- Mejorar la comunicación entre todos.
- Conocer a cada momento que proceso se está ejecutando.

Brechtner (2015) afirma que: “Visualizar el flujo de trabajo es uno de los principios fundamentales de Kanban. Sin comprometerse con el flujo de trabajo de una manera visceral, los miembros del equipo no reconocerán ni resolverán problemas tan rápidamente” (capítulo 1).

Un tablero Kanban cuenta principalmente con 4 columnas, que representan las fases por las que cada tarea tiene que pasar para ser terminada. En cada fase (menos en el backlog) se pasa por dos estados que son “En curso” y “Lista”. El estado de “En curso” significa que actualmente se está trabajando en esa tarea y el estado de “Lista” significa que la tarea se acabó en esa fase y está lista para pasar a la siguiente fase.

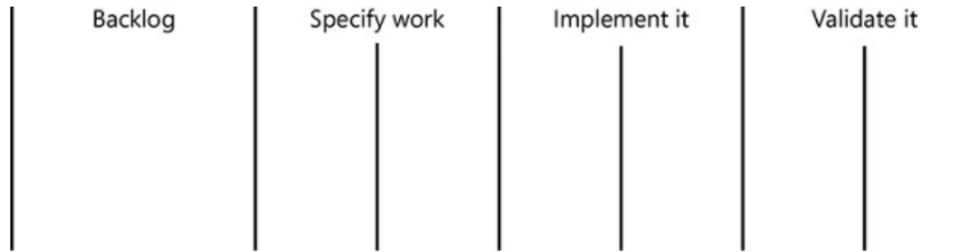


Figura 2: Tablero Kanban

Fuente: (Brechner, E. Agile Project Management with Kanban. 2015)

En la figura 3 se muestra como son colocadas las tareas y cómo van pasando de fase en fase hasta ser entregadas al cliente.

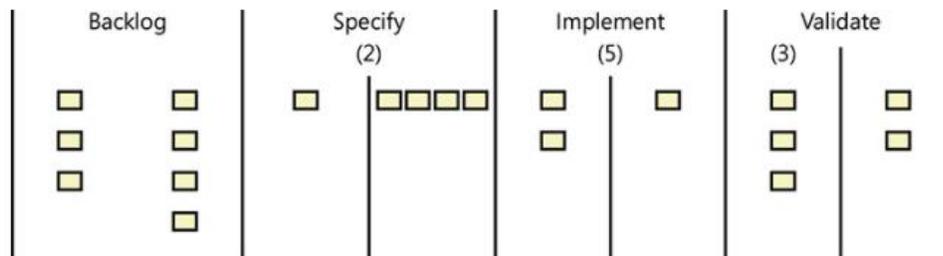


Figura 3: Tablero Kanban con tareas colocadas

Fuente: (Brechner, E. Agile Project Management with Kanban. 2015)

2.7.2.3 Paso 3: Establecer límites al trabajo en curso (WIP)

Una vez que se conoce la realidad del proyecto, se tiene que aplicar una primera restricción en Kanban que es limitar el trabajo en curso (WIP, en inglés work in progress). Se tiene que establecer en cada columna de trabajo las tareas que se pueden realizar.

Al limitar el WIP primero debemos experimentar empíricamente y empezar con un número no muy pequeño de tareas. Lo mejor es empezar con un número alto de tareas para realizar al mismo tiempo y bajar el número de tareas hasta encontrar un equilibrio. (Bermejo, 2015).

“Ajustar el ritmo de los otros pasos para que coincidan produce la mayor eficiencia y la mayor productividad”. (Brechner, 2015).

Los límites de WIP del tablero de trabajo se pueden ajustar en cualquier momento para maximizar el rendimiento y la agilidad del equipo.

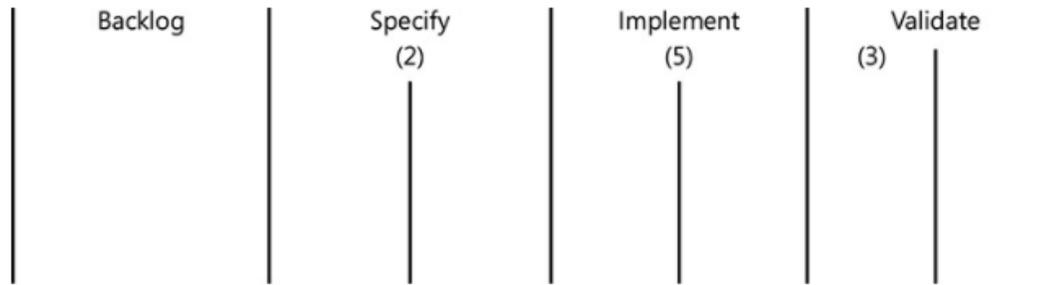


Figura 4: Límites del WIP

Fuente: (Brechtner, E. Agile Project Management with Kanban. 2015)

2.7.2.4 Paso 4: Definir el estado de Hecho

Pasar las tareas de etapa en etapa pareciera ser un mecanismo simple, pero en kanban podemos establecer dos columnas en cada etapa. Se agrega en cada etapa un estado de “En curso” y un estado de “Lista”.

2.7.2.5 Paso 5: Ejecuta tu standup diario

Con las tareas y etapas ya programadas y organizadas no es necesario realizar reuniones de planificación. En Kanban todo fluye continuamente siempre que haya trabajo por hacer.

Se pueden realizar reuniones para revisiones de diseño, demostraciones, reseñas de clientes y discusiones sobre mejoras de productos y procesos.

2.7.3 Métricas en Kanban

Se pueden obtener datos estadísticos del proceso del proyecto.

Una opción es anotar las fechas de entrada y salida de la tarea por cada fase. De esta manera se pueden obtener gráficos del tiempo que tardan las tareas en ser completadas y que fases son las que demoran más en el tiempo.

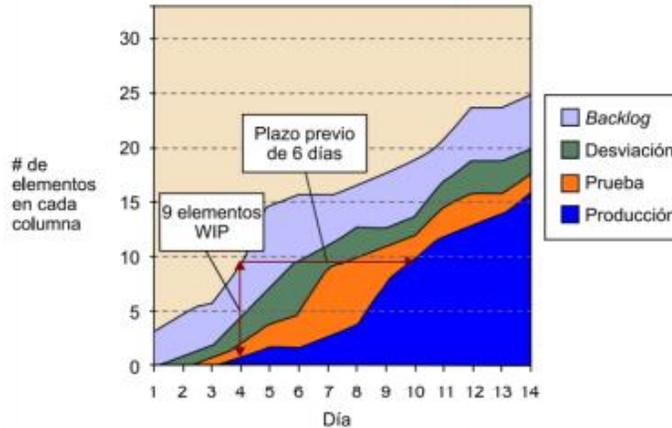


Figura 5: Métricas Kanban
Fuente: (Bermejo, 2015)

2.8 UML

El Lenguaje de modelado unificado (UML) es un lenguaje de modelado estandarizado de uso general. Es ampliamente utilizado y comprendido, lo que lo convierte en una opción popular para modelar una arquitectura de software. (Ingeno, 2018).

Combinar UML y Kanban puede ser complicado si usamos todos los diagramas que UML tiene, Powel (2014) afirma “Aunque la amplitud de notación puede ser un poco abrumadora para los recién llegados, en realidad, un sistema complejo se puede desarrollar fácilmente con tres diagramas centrales: diagramas de clase, diagramas de estado y diagramas de secuencia”. (capítulo 1).

2.8.1 Diagramas de clases

Utiliza diagramas de clases para modelar la vista de diseño estático de un sistema. En su mayor parte, esto implica modelar el vocabulario del sistema, modelar colaboraciones o modelar esquemas. Los diagramas de clases también son la base de un par de diagramas relacionados: diagramas de componentes y diagramas de implementación. (Booch, Rumbaugh y Jacobson, 2005, capítulo 8).

Utiliza diagramas de clases para modelar la vista de diseño estático de un sistema. Esta vista apoya principalmente los requisitos funcionales de un sistema: los servicios que el sistema debe proporcionar a sus usuarios finales.

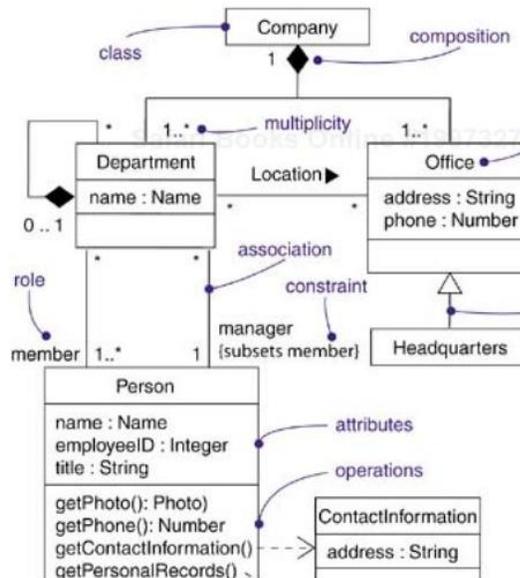


Figura 6: Diagrama de clase
Fuente: (Booch et al, 2015, figura 8-1)

Un diagrama de clases se puede utilizar para:

- modelar una vista de diseño
- modelar vocabulario del sistema
- modelar un esquema de base de datos (Booch 2005).

2.8.2 Diagramas de casos de uso

Los diagramas de casos de uso son fundamentales para modelar el comportamiento de un sistema, un subsistema o una clase. Cada uno muestra un conjunto de casos de uso y actores y sus relaciones.

Los diagramas de casos de uso son importantes para visualizar, especificar y documentar el comportamiento de un elemento. Hacen que los sistemas, subsistemas y clases sean accesibles y comprensibles al presentar una visión externa de cómo se pueden usar esos elementos en contexto. (Booch et al, 2005, capítulo 18).

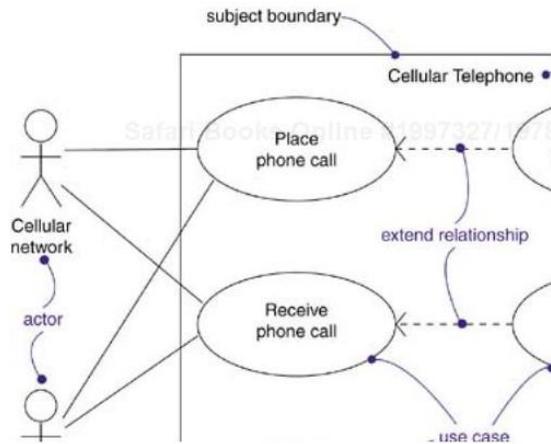


Figura 7: Diagrama de casos de uso
Fuente: (Booch et al, 2015, figura 18-1)

Se aplica el modelado de casos de uso para modelar la vista de casos de uso de un tema y se modelan de dos maneras: Modelar el contexto de un tema y Modelar los requisitos de una asignatura. (Booch et al, 2005, capítulo 18).

2.8.3 Diagramas de interacción

Los diagramas de secuencia y los diagramas de comunicación, que se denominan colectivamente diagramas de interacción, son dos de los diagramas que se utilizan en UML para modelar los aspectos dinámicos de los sistemas. Un diagrama de interacción muestra una interacción, que consiste en un conjunto de objetos y sus relaciones, incluidos los mensajes que se pueden enviar entre ellos. Un diagrama de secuencia es un diagrama de interacción que enfatiza el orden temporal de los mensajes; un diagrama de comunicación es un diagrama de interacción que enfatiza la organización estructural de los objetos que envían y reciben mensajes. (Booch et al, 2005, capítulo 19).

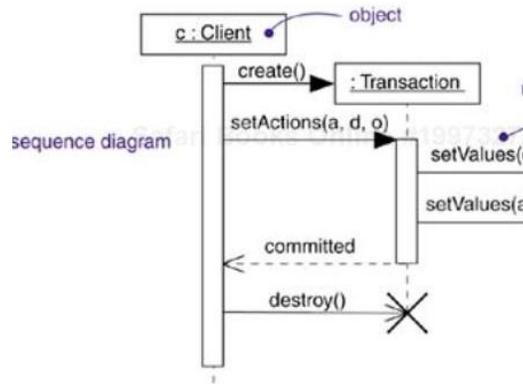


Figura 8: Diagrama de interacción
Fuente: (Booch et al, 2015, figura 19-1)

2.8.2 Diagramas de estado

Los diagramas de estado son uno de los cinco diagramas de UML para modelar los aspectos dinámicos de los sistemas. Un diagrama de estado muestra una máquina de estado. Tanto los diagramas de actividad como los de estado son útiles para modelar la vida útil de un objeto. (Booch et al, 2005, capítulo 25).

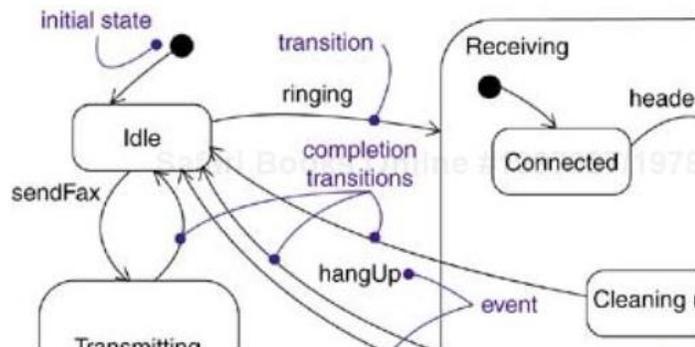


Figura 9: Diagrama de estado
Fuente: (Booch et al, 2015, figura 25-1)

2.9 PATRONES DE DISEÑO

Los patrones de diseño son soluciones simples y elegantes a problemas específicos en el desarrollo de software. Son soluciones que han sido probadas por varios desarrolladores expertos y que han demostrado que funcionan.

La mejor manera de usar patrones es cargar su cerebro con ellos y luego reconocer lugares en sus diseños y aplicaciones existentes donde puede aplicarlos. En lugar

de reutilizar el código, con los patrones obtienes experiencia en la reutilización. (Freeman y Robson, 2020).

Utilizar patrones nos sirve para:

- Proporcionar documentación de elementos reusables en el diseño de sistemas.
- Evitar la reiteración en la búsqueda de soluciones ya conocidos.
- Estandarizar el modo en que se realizar el diseño.
- Facilitar el aprendizaje de nuevas generaciones de diseñadores.

2.9.1 Modelo Vista Controlador

EL Modelo Vista Controlador (MVC), es un patrón de diseño que separa en capas bien definidas el desarrollo de una aplicación, esas partes son tres, el Modelo encargado de la lógica del negocio y la persistencia de los datos, las Vistas son las responsables de mostrar al usuario el resultado que obtienen del modelo a través del controlador, el Controlador encargado es el encargado de gestionar las peticiones del usuario, procesarlas invocando al modelo y mostrarlas al usuario a través de las vistas (Mouse, 2011).

A continuación, se describe cada uno de los componentes citados:

Modelo: El modelo es la representación lógica de la información y describe la funcionalidad del sistema, por lo tanto, se encarga de gestionar los permisos para ingresar a dicha información, también se encarga de realizar recuperación, actualización, inserción y eliminación de datos; a través de un conjunto de reglas (lógica de negocio).

Vista: La vista es la interfaz de usuario. Muestra al usuario una representación visual del modelo, sus datos y estado, tomándolos directamente del modelo. También contiene los elementos de la interfaz que permiten al usuario interactuar con el programa, tales como botones y menús. Sin embargo, no es tarea de la vista implementar cómo se deben comportar esos elementos.

Controlador: El controlador es el intermediario entre la vista y el controlador, responde a eventos generados por el usuario el cual constituyen llamados al Modelo

(en el caso que se solicite alguna información como lista de clientes, consultar un proveedor o editar un producto) o a la Vista (en el caso de mostrar un formulario o un reporte estadístico). En pocas palabras el componente que hace posible tener separada la lógica de negocio con la vista es el Controlador.

2.10 TECNOLOGÍAS WEB

2.10.1 Framework React

React es una biblioteca de *JavaScript* declarativa, eficiente y flexible para crear interfaces de usuario. Le permite componer interfaces de usuario complejas a partir de piezas de código pequeñas y aisladas “componentes”. (Facebook Inc, 2020).

Algunas de las características que tiene este *framework* son:

- **Composición de componentes:** Así como en programación funcional se pasan funciones como parámetro para problemas complejos, en *ReactJS* podemos aplicar este mismo patrón mediante la composición de componentes.
- **Desarrollo declarativo:** En React se cuenta con un estado de la aplicación y sus componentes reaccionan ante el cambio de ese estado.
- **Flujo de datos unidireccional:** En este modelo de funcionamiento, los componentes de orden superior propagan datos a los componentes de orden inferior. Los componentes de orden inferior trabajarán con estos datos y cuando cambia sus estados podrán propagar eventos hacia los componentes de orden superior para actualizar sus estados. Este flujo tiende a ser unidireccional.
- **Performance gracias al DOM Virtual:** El rendimiento de React es muy alto. Esto se consigue por medio del DOM Virtual. No es que React no opere con el DOM real del navegador, pero sus operaciones las realiza antes sobre el DOM Virtual, que es mucho más rápido. El DOM Virtual está cargado en memoria y el resultado es que las operaciones permiten actualizaciones de hasta 60 frames por segundo, logrando así aplicaciones muy fluidas.

- **Elementos y JSX:** El código embebido dentro de JavaScript, parece HTML pero realmente es JSX. Son como funciones JavaScript, pero expresadas mediante una sintaxis propia de React llamada JSX.
- **Los componentes *stateless*:** Son los componentes que no tienen estado, digamos que no guardan en su memoria datos. Eso no quiere decir que no puedan recibir valores de propiedades, pero esas propiedades siempre las llevarán a las vistas sin producir un estado dentro del componente. Estos componentes sin estado se pueden escribir con una sencilla función que retorna el JSX que el componente debe representar en la página.
- **Los componentes *statefull*:** son un poco más complejos, porque son capaces de guardar un estado y mantienen lógica de negocio generalmente. Su principal diferencia es que se escriben en el código de una manera más compleja, generalmente por medio de una clase ES6 (Javascript con ECMAScript 2015), en la que podemos tener atributos y métodos para realizar todo tipo de operaciones. Los componentes *statefull*, con estado, necesitan tener un método `render()` que se encarga de devolver el JSX que usar para representarlo en la página.
- **Ciclo de vida de los componentes:** *React* implementa un ciclo de vida para los componentes. Son métodos que se ejecutan cuando pasan cosas comunes con el componente, que nos permiten suscribir acciones cuando se produce una inicialización, se recibe la devolución de una promesa. (Facebook Inc, 2020).

2.10.2 JavaScript

JavaScript es un lenguaje de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente todo lo demás. (Mozilla, 2020).

El núcleo del lenguaje JavaScript de lado del cliente consta de algunas características de programación comunes que te permiten hacer cosas como:

- Almacenar valores útiles dentro de variables.

- Operaciones sobre fragmentos de texto (conocidas como "cadenas" (*strings*) en programación).
- Ejecuta código en respuesta a ciertos eventos que ocurren en una página web. (Mozilla, 2020).

2.10.3 Backend como un servicio (BaaS Backend as a Service)

“*Backend as a Service* (BaaS), que tiene sus raíces en *Mobile Backend as a Service* (MBaaS), es un modelo que permite a los desarrolladores aprovechar las aplicaciones de servicio proporcionadas por terceros. Esto reduce el tiempo y los costos de desarrollo, ya que los equipos no tienen que escribir estos servicios internamente. (...), BaaS ofrece el uso de servicios existentes”. (Firebase, 2020).

Estos servicios de BaaS casi siempre incluyen:

- Una base de datos
- Notificaciones push
- Almacenamiento de archivos
- Servicios de autenticación.

Esto cuenta con la ventaja que el desarrollo de software se simplifica en gran parte por que la lógica detrás del servidor esta ya pre elaborada y disponible para su implementación en el proyecto.

2.10.3.1 Cloud Firestore

Cloud Firestore es una base de datos *NoSql* que está estructurada para ser utilizada mediante la construcción de colecciones, subcolecciones y documentos, los accesos a los datos se hacen mediante consultas o peticiones a través del SDK de *Firebase*.

Cloud Firestore es una base de datos flexible y escalable para la programación en servidores, dispositivos móviles y la Web desde *Firebase* y *Google Cloud Platform*.

Al igual que *Firebase Realtime Database*, mantiene tus datos sincronizados entre apps cliente a través de agentes de escucha en tiempo real y ofrece asistencia sin conexión para dispositivos móviles y la Web, por lo que puedes compilar apps con

capacidad de respuesta que funcionan sin importar la latencia de la red ni la conectividad a Internet. Cloud *Firestore* también ofrece una integración sin interrupciones con otros productos de *Firebase* y *Google Cloud Platform*, incluido *Cloud Functions*. (Firebase, 2020).

Las funciones más importantes en esta herramienta son:

- **Flexibilidad:** Admite estructuras de datos flexibles y jerárquicas.
- **Consultas expresivas:** Las consultas a la base de datos pueden incluir varios filtros en cadena y combinar filtros de criterios de orden.
- **Actualizaciones en tiempo real:** Se usa la sincronización de datos para actualizar los datos de cualquier dispositivo conectado.
- **Asistencia sin conexión:** Los datos que usa el sistema se almacenan en caché de datos.
- **Diseñado para ajustarse a escala:** Se tiene una infraestructura con replicación automática de datos multirregión, garantías de coherencia sólida.

2.10.3 Bases de Datos NoSQL

Hoy en día, hay cientos de bases de datos NoSQL disponibles en el mundo, y algunas populares, como MongoDB, HBase y Cassandra, tienen la mayor parte del mercado, seguidas de una larga lista de otras bases de datos menos populares.

Estas bases de datos tienen diferentes modelos de datos, que van desde el modelo de documento de MongoDB, al modelo de familia de columnas de HBase y Cassandra, al modelo de columnas de Kudu. Estas bases de datos se implementan ampliamente en cientos de organizaciones y, en este punto, se consideran convencionales y comunes. (Ploetz, Kandhare, Kadambi y Wu, 2018, introducción a NoSQL).

Las bases de datos NoSQL admiten una variedad de modelos de coherencia, como los siguientes:

- **Fuerte consistencia:** un sistema que es fuertemente consistente asegura que las actualizaciones de una clave determinada estén ordenadas y las lecturas reflejen la última actualización que ha sido aceptada por el sistema.

- **Coherencia de la línea de tiempo:** un sistema que es coherente con la línea de tiempo garantiza que las actualizaciones de una clave determinada se ordenen en todos los replicantes, pero las lecturas en un replicante determinado pueden estar obsoletas y es posible que no reflejen la última actualización que ha aceptado el sistema
- **Consistencia eventual:** un sistema que es eventualmente consistente no garantiza si las actualizaciones se aplicarán en orden en todos los replicantes, ni ofrece garantías sobre cuándo una lectura reflejaría una actualización previa aceptada por el sistema. (Ploetz et al ,2018, introducción a NoSQL).

2.11 LEARNING MANAGMENT SYSTEM (LMS)

Un sistema de gestión del aprendizaje (LMS) no es el remedio para todos los desafíos de aprendizaje de su organización. Pero puede ser útil, especialmente para organizaciones que brindan programas de capacitación y educación a empleados, clientes, proveedores, distribuidores, miembros de asociaciones profesionales, personal militar, estudiantes universitarios y otros. Si su organización es una de estas, probablemente sepa que administrar la entrega de programas de aprendizaje presenta muchos desafíos. (Foreman, 2017).

Un LMS es una aplicación de software multiusuario, a la que generalmente se accede a través de un navegador web. Ayuda a las organizaciones a gestionar eventos de formación, cursos a su propio ritmo y programas de aprendizaje mixto. Proporciona automatización que reemplaza el trabajo manual riguroso y costoso, ahorra tiempo y le permite organizar su contenido, datos y alumnos. Realiza un seguimiento e informa sobre la actividad y los resultados de la formación. (Foreman, 2017).

2.12 METODOLOGÍA WEBQEM

La metodología WebQem toma métricas del modelo de calidad ISO 9126-1 la cual da referencia a las siguientes características, funcionalidad, confiabilidad, usabilidad, mantenibilidad y portabilidad.

Funcionalidad: Grado en el que el software satisface las necesidades planteadas según los subatributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.

La funcionalidad es una métrica orientada a la función del sistema y al proceso por el cual se desarrolla. Se centra en la funcionalidad o utilidad del programa.

- Entradas del usuario, se toma en cuenta cada entrada del usuario que el sistema proporciona a medida que ingresa al sistema.
- Salidas del usuario, se refleja las salidas que tiene el sistema tanto reportes como estadísticas que tiene el sistema.
- Número de peticiones del usuario, una petición se define como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida.
- Número de archivos, se define cada archivo lógico.
- Número de interfaces externas, se definen todas aquellas interfaces legibles por el ordenador que solicitan transmitir información a otro sistema.

Para calcular los puntos función se usa las siguientes formulas:

$$PF = Cuenta\ Total * (confiabilidad\ proyecto + error\ min * \sum Fi) \quad (1)$$

$$Funcionalidad = \left(\frac{PF}{PF_{m\acute{a}ximo}} \right) \quad (2)$$

Donde:

PF: Medida de Funcionalidad.

PF_{máximo}: Medida de funcionalidad con su valor máximo.

Cuenta total: La suma de Número de entradas, número de salidas, número de peticiones, número de archivos y número de interfaces externas.

Confiabilidad proyecto: Varía de 1% al 100% (0 a 1).

Error min: Error mínimo aceptable de complejidad.

$\sum Fi$: Son los valores de ajuste de complejidad, donde ($1 \leq i \leq 14$).

Confiabilidad: Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes subatributos: madurez, tolerancia a fallas y recuperación.

Es la probabilidad de operación libre de fallos en un programa en un entorno determinado y durante un tiempo específico se toma en cuenta:

a) Enlaces

- Enlaces rotos
- Enlaces inválidos
- Enlaces no Implementados

b) Páginas

- Páginas muertas
- Páginas bajo Construcción
- Errores de Ortografía

Para calcular los puntos de función se usa la siguiente fórmula:

$$\text{Porcentaje de enlaces rotos} = \frac{CERI+CERE}{CTE} * 100 \quad (3)$$

Donde:

CERI: Cantidad de enlaces rotos internos

CERE: Cantidad de enlaces rotos externos

CTE: Cantidad total de enlaces

Porcentaje de presencia de propiedad:

$$\text{Porcentaje presencia ALT} = \frac{\text{CantidadImágenesALT}}{\text{CantidadTotalImágenes}} * 100 \quad (4)$$

La confiabilidad se calculará con la siguiente fórmula:

$$\text{Confiabilidad} = \text{PorcentajeEnlaces} + \text{PorcentajePresenciaALT} \quad (5)$$

Usabilidad: Grado en el que el software es fácil de usar, según los siguientes atributos: Comprensibilidad, Mecanismo de ayuda, Aspecto de interfaces, Aspectos estéticos y de estilo.

Toma en cuenta la capacidad del software para ser comprendido, utilizado y atractivo para el usuario en determinadas condiciones.

Se realiza una tabla que toma en cuenta los siguientes puntos:

a) Comprensión Global del sitio

- Lista de contenidos
- Mapa del sitio
- Índices
- Visita guiada
- Mapa de imagen

b) Aspectos de interfaces y estéticos

- Controles directos
- Controles indirectos
- Estabilidad

c) Mantenimiento del color de los enlaces

Se toma en cuenta el diseño que tiene el sistema

Eficiencia: Grado en el que el software emplea óptimamente los recursos del sistema, según sus subatributos: comportamiento del tiempo y de los recursos.

Mantenibilidad: Facilidad con la que pueden efectuarse reparaciones al software, según sus subatributos: analizable, cambiable, estable, susceptible de someterse a pruebas.

El índice de madurez del software se calcula con la siguiente formula:

$$IMS = \frac{[Mt - (Fa + Fb + Fc)]}{Mt} \quad (6)$$

Donde:

Mt: Número de módulos en la versión actual

Fa: Número de módulos en la versión actual que se han cambiado

Fb: Número de módulos en la versión actual que se han añadido

Fc: Número de módulos en la versión anterior que se han borrado en la versión actual.

Portabilidad: Facilidad con la que el software puede llevarse de un ambiente a otro según lo indican sus subatributos: adaptable, instalable, conformidad(coexistencia) y sustituible (Pressman, 2010).

Para poder medir el grado de portabilidad usaremos la siguiente fórmula:

$$GP = 1 - \frac{ET}{ER} \quad (7)$$

Donde:

ET: Es la medida de los recursos necesarios para llevar el sistema a otro entorno.

ER: Es la medida de los recursos necesarios para crear el sistema en el entorno residente.

Si $GP > 0$, la portabilidad es más rentable que el re-desarrollo

Si $GP = 1$, la portabilidad es perfecta

Si $GP < 0$, el re-desarrollo es más rentable que la portabilidad

A continuación, se describe las fases de la metodología WebQem para medir la calidad del software.

2.12.1 Definición de las metas de evaluación y selección del perfil de usuario

En esta fase se consideran dos pasos primordiales y comunes en toda evaluación de calidad siguiendo la metodología WebQem.

a) Metas de Evaluación

En esta fase se define las metas de evaluación y selección del perfil de usuario, los evaluadores deben definir las metas y establecer el alcance del proyecto.

Las metas a llegar a cumplir con la evaluación del sistema son:

- Conocer la percepción de la calidad que tienen los usuarios finales sobre el sistema.

- Lograr que la calidad del sistema esté por encima de las expectativas del usuario final.

b) Selección de Perfil de Usuario

Para el diseño e implementación de un caso de estudio de evaluación de calidad en uso, una meta muy utilizada es determinar el cumplimiento de requerimientos elementales, parciales y globales de calidad para una aplicación Web.

2.12.2 Definición de criterios elementales y procedimientos de medición

En esta etapa se define una base de criterios para evaluación elemental, y realizar el proceso de medición y puntuación de los valores que se calcule para medir la calidad del software en tal caso se definirá los siguientes criterios:

- Mala: 0-30%
- Regular: 31 – 50%
- Bueno: 51 – 90%
- Muy bueno: 91 – 100%

2.12.3 Definición de agregación e implementación de la evaluación global

Para esta fase se realiza un promedio de todos los parámetros encontrados para poder sacar un resultado global y así el usuario pueda comprender de mejor manera la calidad del sistema.

2.12.4 Análisis de resultados y recomendaciones

En esta fase una vez calculado las características de calidad se dan las recomendaciones respectivas además de saber los resultados y ver si el sistema cumple con lo pronosticado.

2.13 COCOMO

El modelo Cocomo (Constructive Cost Model) desarrollado por Barry M. Boehm, se centra en modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto.

Cocomo define tres modos de desarrollo:

Orgánico: Proyectos sencillos, que tienen menos de 5000 líneas de código, en donde se tiene experiencia de proyectos similares.

Semi-acoplado: Proyectos intermedios en complejidad y tamaño, donde la experiencia en el proyecto es variable.

Empotrado: Proyectos bastante complejos, donde la experiencia es mínima y el entorno es de innovación técnica.

Los Modelos que define COCOMO:

Modelo básico: Se basa en el tamaño de líneas de código (LDC).

Modelo intermedio: Se basa en el tamaño del programa y el conjunto de medidas subjetivas llamadas conductores de costes.

Modelo avanzado: Se incluye el modelo intermedio y el impacto de cada conductor de coste en todas las fases de desarrollo.

Para el proyecto de grado se utiliza el modelo intermedio para realizar las estimaciones de costes.

Las formulas son las siguiente:

$$\bullet E = \text{Esfuerzo} = a \text{KLDC} e * \text{FAE} \left(\frac{\text{persona}}{\text{mes}} \right) \quad (7)$$

$$\bullet T = \text{T tiempo de duración del desarrollo} = c \text{Esfuerzo} d(\text{mes}) \quad (8)$$

$$\bullet P = \text{Personal} = \frac{E}{T} (\text{personas}) \quad (9)$$

2.14 CASOS DE PRUEBA

Para realizar las pruebas es necesario la creación de casos de prueba especificando la forma de probar el sistema como un todo. (Sommerville, 2005).

- Realizar pruebas en la plataforma definida.
- Pruebas de configuración.
- Pruebas negativas.
- Pruebas de tensión o estrés.
- Prueba de integración del sistema. (Sommerville, 2005).

2.15 PRUEBAS UNITARIAS

Es parte de los principios de preparación que se debe de considerar antes de escribir una sola línea de código. Se debe de crear un conjunto de pruebas que se aplicarán una vez que se haya terminado el componente a codificar. (Pressman, 2010, p. 94).

Las pruebas unitarias por es preparadas para cada componente serán utilizadas para el presente proyecto por su naturaleza compatible con el *framework* de desarrollo *React*.

2.16 SEGURIDAD

2.16.1 Principios de Seguridad

La seguridad del software es una actividad del aseguramiento del software que se centra en la identificación y evaluación de los peligros potenciales que podrían afectarlo negativamente y que podrían ocasionar que falle todo el sistema. Si los peligros se identifican al principio del proceso del software, las características de su diseño se especifican de modo que los eliminen o controlen.

La seguridad en informática es la capacidad de ofrecer confidencialidad, integridad, disponibilidad y control de la información conocida solo por los individuos autorizados (Pressman, 2010, p. 378)

2.16.2 Seguridad Lógica

Consiste en la aplicación de barreras y procedimientos que resguarden el acceso a datos teniendo un control de las personas autorizadas, (Pressman, 2010, p. 378)

Entre los objetivos de la seguridad lógica están las restricciones de acceso a los programas y archivos, supervisión en el trabajo de los operadores para evitar la modificación del código del sistema, asegurar que se usan los archivos, programas y datos correctos y por último asegurarse de que la información enviada sea igual a la recibida. (Pressman, 2010, p. 378).

La seguridad lógica hace referencia a la aplicación de mecanismos y barreras para mantener el resguardo y la integridad de la información dentro de un sistema

informático La seguridad lógica se complementa con la seguridad física, por lo general se divide, en cinco objetivos principales.

- Integridad: garantizar que los datos sean los que se supone que son
- Confidencialidad: asegurar que sólo los individuos autorizados tengan acceso a los recursos que se intercambian
- Disponibilidad: garantizar el correcto funcionamiento de los sistemas de información
- Autenticación: asegurar que sólo los individuos autorizados tengan acceso a los recursos

2.16.3 Seguridad Física

Mecanismos de prevención y detección para la protección física de los recursos del sistema. Si un individuo desea atacar un sistema y tiene acceso físico este, toda medida de seguridad implementada es inútil. Por esto es conveniente el control de acceso a salas y protegerlas por medios biométricos, llaves entre otros, (Pressman, 2010, p. 379)

CAPITULO III

3. DISEÑO METODOLÓGICO

3.1 INTRODUCCIÓN

En este capítulo se empieza con una descripción de la forma de trabajo combinando la metodología Kanban con UML.

También se hace una descripción de la forma de trabajo realizado en la institución realizado por el personal administrativo, docente y estudiantes.

Se describe también las fases de desarrollo de los diferentes módulos del proyecto utilizando los diagramas de UML.

3.2 ANALISIS DE LA INSTITUCIÓN

Los trabajos de mayor demanda en la institución en un inicio son:

- La inscripción de estudiantes a los distintos cursos del nivel secundario.
- Designación de docentes para los distintos cursos.
- Organización del horario escolar para la gestión.
- Planificación de contenidos curriculares

En cuanto al desarrollo de actividades en la institución podemos mencionar las más sobresalientes:

- El desarrollo de las clases en la modalidad virtual se da en los horarios previamente establecidos y comunicados por el personal administrativo.
- Cada docente lleva un registro de todas las actividades y contenidos desarrollados con cada curso.
- Cada docente elige las herramientas y plataformas que le ayuden a desarrollar las clases con los estudiantes.
- El docente tiene la responsabilidad de evaluar cada trabajo llevado a cabo por los estudiantes tomando en cuenta criterios de evaluación.
- Los contenidos curriculares son organizados en tres trimestres

- En cada trimestre cada docente debe de realizar el informe de las actividades realizadas y así mismo de dar una calificación trimestral a cada estudiante.
- La evaluación de los docentes esta normada y establecida por el Ministerio de Educación
- Todos los documentos son centralizados por el personal administrativo.

3.3 COBINACIÓN DE KANBAN CON UML

Con Kanban podremos visualizar el trabajo que se está realizando y el trabajo que está listo, esto ayuda a determinar en qué fases tenemos cuellos de botella y que fases están sin desarrollar aún. Así mismo podremos realizar métricas para ajustar los tiempos de desarrollo.

El tablero Kanban que utilizaremos tendrá cuatro fases de desarrollo:

- **Pedidos:** En esta fase se representan todos los requerimientos que se necesitan para el desarrollo del sistema.
- **Especificación:** En esta fase se analiza las tareas necesarias para cumplir el pedido, si el trabajo es muy grande, entonces se divide en pequeñas tareas de igual tamaño y se plantea un diseño.
- **Implementación:** En esta fase se realiza el desarrollo del sistema y la implementación.
- **Validación:** En esta fase se realizan pruebas al sistema según reglas de validación establecidos previamente.

3.4. KANBAN ROLES

Tabla 1 Fases de Trabajo, WIP y responsables

Fase	WIP	Responsable
Pedidos	-	Propietario del Producto: Lic. Wilfredo Villegas Alvarez David Gonzalo Cusi Quispe
Especificación	3	Analista: David Gonzalo Cusi Quispe
Implementación	2	Diseñador y Desarrollador: David Gonzalo Cusi Quispe
Validación	2	Desarrollador: David Gonzalo Cusi Quispe

3.4 MÓDULO DE ADMINISTRACIÓN DE USUARIOS

3.4.1 Fase de Pedidos

En ésta fase se detalla la lista de los actores involucrados para el sistema:

- El Director y secretarios tendrán las opciones de crear, editar, ver y eliminar su cuenta y las cuentas de docentes y estudiantes.
- El Director y Secretarios tendrán la opción de generar las cuentas de todos los usuarios docentes y estudiantes.
- Los docentes podrán crear, editar, ver y eliminar su cuenta.
- Los estudiantes podrán crear, editar, ver y eliminar su cuenta.

3.4.1.1 Tablero Kanban

Según los requerimientos obtenidos se necesita un Módulo de Administración de Usuarios marcado en el tablero Kanban con el color amarillo.



Figura 10: Tablero Kanban columna de pedidos

3.4.2 Fase de Especificación

3.4.2.1 Casos de uso: Administrar Usuarios

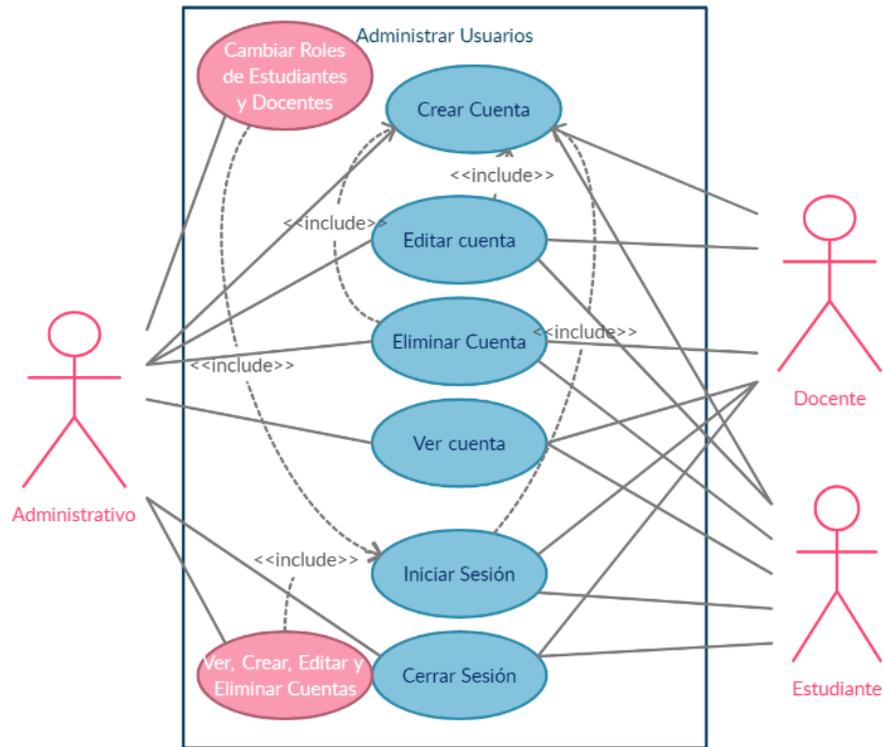


Figura 11: Casos de uso: Administrar Roles de Usuario

Se detalla las especificaciones de los casos de uso Administrar Usuarios:

Tabla 2: Especificación del caso de uso Crear Cuenta

Nombre	Crear Cuenta
Descripción	Permite a los usuarios crear cuentas en el sistema
Actores	Personal Administrativo, Docente y Estudiante
Precondiciones	El actor debe de contar con la clave de acceso valida del sistema.
Postcondiciones	Se cuenta con una cuenta creada y un rol asignado en el sistema.
Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema. 2. El usuario proporciona los datos de registro para el sistema incluida la clave de acceso. 3. El usuario realiza la petición de crear cuenta.

	<ol style="list-style-type: none"> 4. El sistema verifica si el código de acceso es válido. 5. El sistema crea una cuenta con un rol asignado de acuerdo al código de acceso. 6. El sistema muestra el mensaje de bienvenida.
Flujo Alternativo	<p>Datos incompletos</p> <p>En el paso 3 si los datos requeridos están incompletos el sistema muestra un mensaje de error.</p> <p>Usuario repetido</p> <p>En el paso 3 si la cuenta ya ha sido creada anteriormente el sistema muestra un mensaje de error.</p> <p>Código de acceso no valido</p> <p>En el paso 4 si el código de acceso es invalido el sistema muestra un mensaje de error.</p>

Tabla 3: Especificación de casos de uso Editar Cuenta y Eliminar Cuenta

Nombres	Editar Cuenta y Eliminar Cuenta
Descripción	Permite a los usuarios Editar y eliminar sus cuentas en el sistema
Actores	Personal Administrativo, Docente y Estudiante
Precondiciones	El Actor debe tener una cuenta creada en el sistema
Postcondiciones	El Actor tiene cambios en su cuenta o el actor ya no tiene acceso al sistema.
Flujo Normal	<ol style="list-style-type: none"> 1. El actor ingresa al sistema. 2. Se inicia sesión con una cuenta valida. 3. El actor realiza la petición de ver Perfil de la cuenta. 4. El actor modifica o elimina su cuenta de usuario. 5. El sistema guarda los cambios en la base de datos y muestra un mensaje de la operación.

Flujo Alterno	<p>Datos incompletos</p> <p>En el paso 4 si los datos requeridos están incompletos el sistema muestra un mensaje de error.</p> <p>Cuenta no Valida</p> <p>En el paso 2 si la cuenta no es válida en el sistema, se muestra un mensaje de error</p>
----------------------	--

Tabla 4: Especificación del caso de Uso Cambiar Roles de Estudiantes y Docentes

Nombre	Cambiar Roles de Estudiantes y Docentes
Descripción	Permite al actor cambiar los roles de estudiantes y docentes
Actores	Personal Administrativo
Precondiciones	El Actor debe tener una cuenta creada en el sistema con el rol de administrador
Postcondiciones	Se actualizan los roles de usuarios del sistema
Flujo Normal	<ol style="list-style-type: none"> 1. El actor administrativo ingresa al sistema. 2. Se inicia sesión con una cuenta valida. 3. El actor realizar la petición de ver lista de Usuarios. 4. El actor modifica los roles de usuario. 5. El sistema guarda los cambios en la base de datos y muestra un mensaje de la operación.
Flujo Alterno	<p>Operación de escritura fallida</p> <p>En el paso 5, si existe un fallo el sistema muestra un mensaje de error.</p>

Tabla 5: Especificación del caso de uso Ver, Crear, Editar y Eliminar Cuentas

Nombre	Ver, Crear, Editar y Eliminar Cuentas
Descripción	Permite al actor realizar operaciones de lectura y escritura en las cuentas de usuarios

Actores	Personal Administrativo
Precondiciones	El Actor debe tener una cuenta creada en el sistema con el rol de administrador
Postcondiciones	Se actualizan los roles de usuarios del sistema
Flujo Normal	<ol style="list-style-type: none"> 1. El actor administrativo ingresa al sistema. 2. Se inicia sesión con una cuenta valida. 3. El actor realizar la petición de ver lista de Usuarios. 4. El actor modifica las cuentas de usuarios del sistema. 5. El sistema guarda los cambios en la base de datos y muestra un mensaje de la operación.
Flujo Alternativo	<p>Operación de escritura fallida</p> <p>En el paso 5, si existe un fallo el sistema muestra un mensaje de error.</p>

3.4.2.2 Diagrama de Clases: Administrar Usuarios

En el diagrama de clases se muestra los datos que son guardados de los usuarios. Se guarda en la **colección UsuariosFirebase** los datos ingresados de todos los usuarios y en la **colección users** se guardan los roles e información de los usuarios.

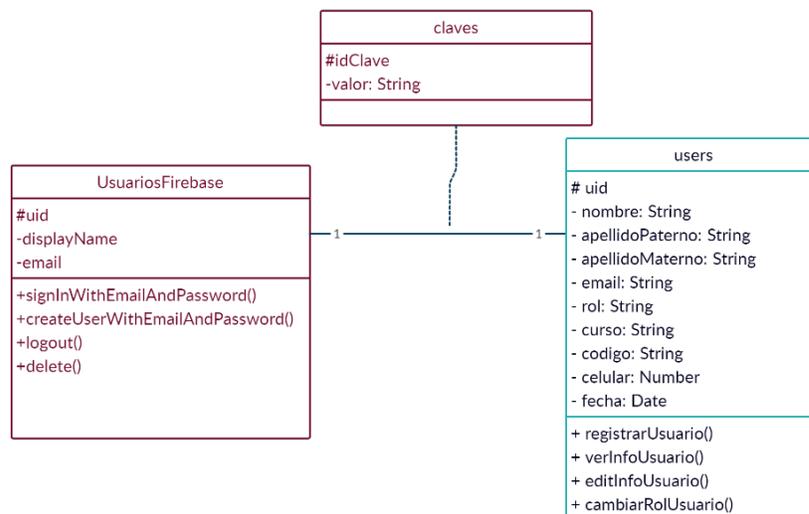


Figura 12: Diagrama de Clases: UsuariosFirebase y users

3.4.2.3 Tablero Kanban

Al ser el Módulo de Administración de Usuarios parte principal del sistema se especifica y divide en tareas. Esta división de tareas se muestra en la siguiente figura.

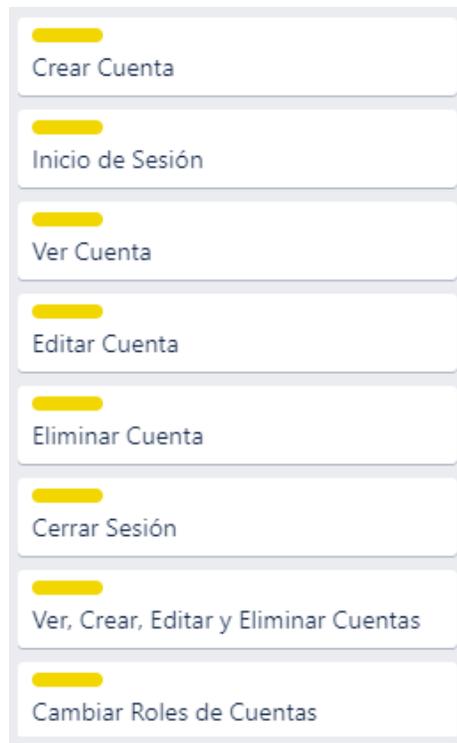


Figura 13: Tablero Kanban columna de especificación

3.4.3 Fase de Implementación

En la fase de implementación se trabajará considerando el WIP de 2 tareas.

3.4.3.1 Diagrama de estado: Administrar Usuarios

El diagrama de estado muestra las páginas y su comportamiento accediendo al sistema. El usuario al iniciar sesión se encontrará con la página Home y la Barra de Navegación en donde se le mostrarán opciones de usuario dependiendo de su rol y si el usuario no cuenta con un rol válido el sistema lo redirigirá a una página con un mensaje de aclaración.

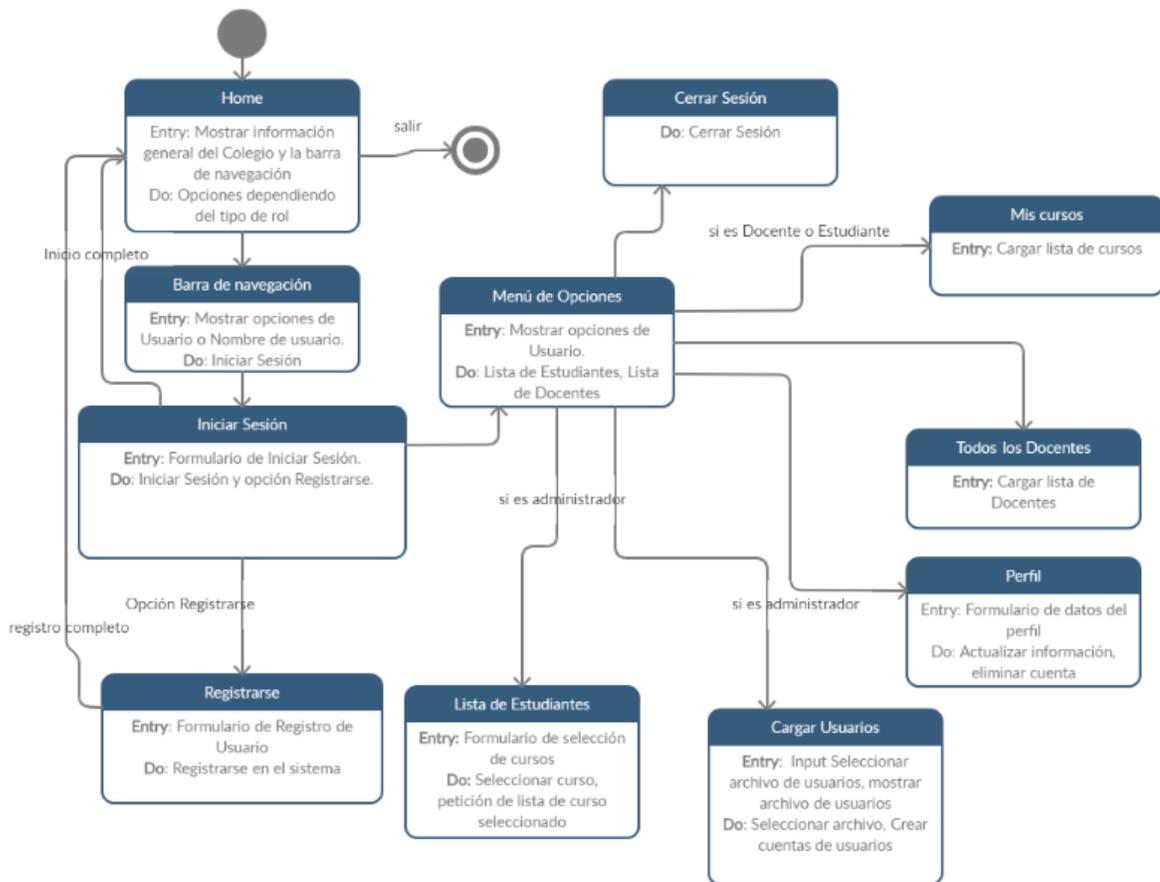


Figura 14: Diagrama de estado Administración de Usuarios

3.4.3.2 Tarea: Crear cuenta

En la figura se muestra la interfaz para crear cuenta de usuario. La interfaz es accedida creando el componente CrearCuenta en React. Disponible solo con un código de acceso valido.

U.E. "Calama"
Aplicación Web

Iniciar Sesión

Bienvenido

Por favor proporcione los siguientes detalles de su nueva cuenta

Codigo de acceso

Nombres

Apellido Paterno

Apellido Materno

Curso

Selecciona un Curso

Celular

Correo Electronico Gmail

Contraseña

Registrar mi cuenta

[¿Ya tienes una cuenta?-Inicia Sesión](#)

Figura 15: Interfaz Crear Cuenta

3.4.3.3 Tarea: Inicio de sesión

En la figura se muestra la interfaz para iniciar sesión con una cuenta creada. También se crea un componente de Inicio de Sesión utilizando la librería de React.

U.E. "Calama"
Aplicación Web

Iniciar Sesión

Bienvenido!

Inicia sesión para continuar

Correo Electronico Gmail

Contraseña

[¿Olvidaste tu contraseña?](#)

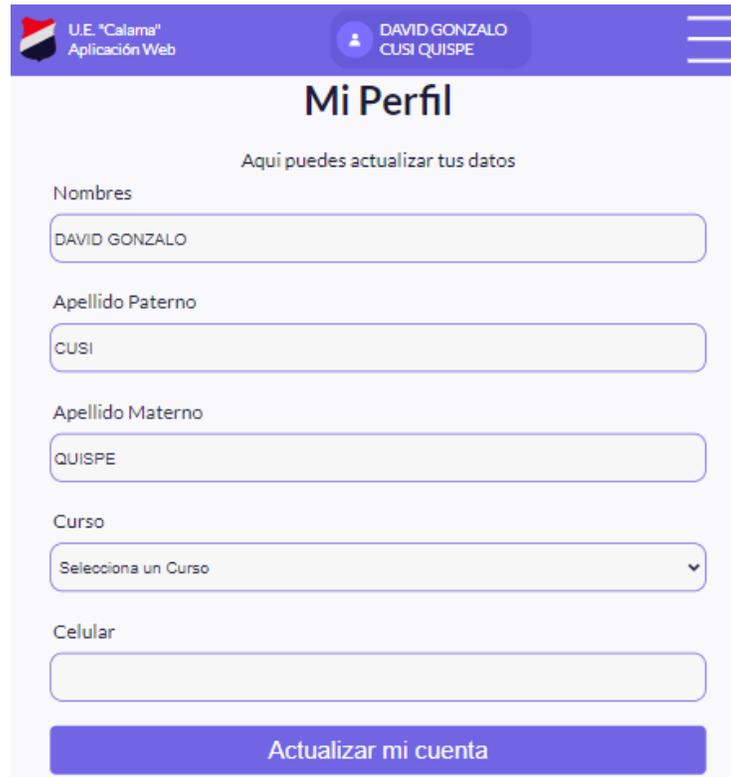
Iniciar Sesión

[¿No tienes una cuenta?-Regístrate](#)

Figura 16: Interfaz Inicio de Sesión

3.4.3.4 Tarea: Editar Cuenta

En la figura se muestra la interfaz para Editar la cuenta una vez se haya iniciado sesión. También se crea un componente *EditarCuenta* utilizando la librería de *React*. Solo serán accedido por los usuarios que hayan iniciado sesión.



The screenshot shows a web application interface for editing a user profile. At the top, there is a purple header bar containing the logo of U.E. 'Calama' (Universidad Especializada de Calama) on the left, the user's name 'DAVID GONZALO CUSI QUISPE' in the center, and a hamburger menu icon on the right. Below the header, the main content area is titled 'Mi Perfil' (My Profile) in a large, bold font. Underneath the title, there is a subtitle 'Aquí puedes actualizar tus datos' (Here you can update your data). The form consists of several input fields: 'Nombres' (Names) with the value 'DAVID GONZALO', 'Apellido Paterno' (Paternal Surname) with 'CUSI', 'Apellido Materno' (Maternal Surname) with 'QUISPE', 'Curso' (Course) with a dropdown menu showing 'Selecciona un Curso', and 'Celular' (Cellular) which is currently empty. At the bottom of the form, there is a prominent purple button labeled 'Actualizar mi cuenta' (Update my account).

Figura 17: Interfaz Editar Cuenta

3.4.3.5 Tarea: Ver, Crear, Editar y Eliminar Cuentas

En la figura se muestra la interfaz para Editar la cuenta una vez se haya iniciado sesión. También se crea un componente *EditarCuenta* utilizando la librería de *React*. La interfaz solo será habilitada para usuarios registrados con el rol de administrador.

U.E. "Calama" Aplicación Web		DAVID GONZALO CUSI QUISPE	
Cargar Usuarios <input type="button" value="Seleccionar archivo"/> segundo_a.xlsx			
1	LIDIA APAZA MITA	Curso 2 Segundo A	Correo apazamitalidia@gmail.com
2	JACQUELINE JENNY APAZA TUMIRI	Curso 2 Segundo A	Correo apazatumirijacquelinejenny@gmail.com
3	LIZETH ANABEL AYMA GUTIERREZ	Curso 2 Segundo A	Correo aymagutierrezlizethanabel@gmail.com
4	ALEX JOSE CANAVIRI CANDIA	Curso 2 Segundo A	Correo canaviricandiaalexjose@gmail.com
5	YESENIA CHINO CONDORI	Curso 2 Segundo A	Correo chinocondoriyesenia@gmail.com
6	LILIAM ADELA CHOQUE CRUZ	Curso 2 Segundo A	Correo choquecruzliliamadela@gmail.com
7	ZEYNA WARA CUSI ORTIZ	Curso 2 Segundo A	Correo cusiortizzeynawara@gmail.com
8	PAULO ANDRE FLORES QUECAÑA	Curso 2 Segundo A	Correo floresquecanapauloandre@gmail.com

Figura 18: Interfaz Editar y Eliminar cuentas

3.4.4 Fase de Validación

En esta fase se determinan reglas de validación para cada tarea completada.

- Funcionamiento correcto en local
- Funcionamiento correcto con la aplicación desplegada
- Funcionamiento en dispositivos móviles, escritorio y navegadores

Realizamos la siguiente tabla para anotar las validaciones del módulo administración de usuarios.

Tabla 6: Validación del Módulo Administración de Usuarios

Validación Administración de Usuarios	
Validaciones	Estado(SI/NO)
Revisión del Código	SI
Verificar pruebas unitarias	SI

Verificación de la documentación	SI
Integración del Módulo al repositorio	SI
Despliegue de código a servidor	SI

3.4.4.1 Pruebas unitarias

En esta fase se realiza las pruebas unitarias a cada submodulo del sistema estas pruebas se realizan con la herramienta Jest que viene pre configurada en el proyecto react.

En la siguiente tabla se muestra las siguientes pruebas unitarias realizadas en los componentes del módulo de administración de usuarios.

Tabla 7: Pruebas del Módulo Administración de Usuarios

Pruebas	Resultado
El sistema registra usuarios que no proporcionen todos los datos	Cumple
El sistema no registra usuarios que tengan campo mal llenados	Cumple
El componente de Crear Cuenta se muestra correctamente	Cumple
El componente de Iniciar Sesión se muestra correctamente	Cumple

En la siguiente figura se muestran los resultados obtenidos en la consola que demuestran el correcto funcionamiento de los dos componentes de forma individual.

```

PASS src/__test__/components/CrearCuenta.test.js (8.355s)
  Tests de modulo administracion de servicios
    ✓ Render del componente CrearCuenta (33ms)
    ✓ Render del componente IniciarCuenta (15ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        8.452s
Ran all test suites matching /src\/__test__\/components\/CrearCuenta.test.js/i.

Watch Usage: Press w to show more.

```

Figura 19: Resultado de pruebas unitarias módulo de administración de usuarios

3.5 MÓDULO DE SERVICIOS

3.5.1 Fase de Pedidos

Detallamos la lista de pedidos sobre las operaciones de los usuarios:

- Personal Administrativo son los encargados de publicar Horarios, comunicados, cambiar la información de los estudiantes, cambiar de curso a los estudiantes, gestionar las cuentas de estudiantes y docentes,
- Los docentes son los encargados gestionar los cursos y materias organizados por trimestre, desarrollar contenidos tareas y actividades en entornos virtuales y revisar las tareas y actividades realizadas de cada estudiante.
- Los docentes tienen la responsabilidad de asignar calificaciones y evaluar a los estudiantes con los criterios y dimensiones establecidos por el ministerio de educación los cuales son: Saber, Hacer, Ser/Decidir. Para este pedido se analiza el cuadro de evaluación con el que actualmente trabajan los docentes en el Anexo G.
- Los estudiantes son los responsables de ver los cursos y contenidos y participar mediante la entrega de tareas, participaciones, preguntas y respuestas sobre los contenidos llevados por el docente.

3.5.2 Fase de Especificación

3.5.2.1 Casos de Uso: Operaciones del Administrativo

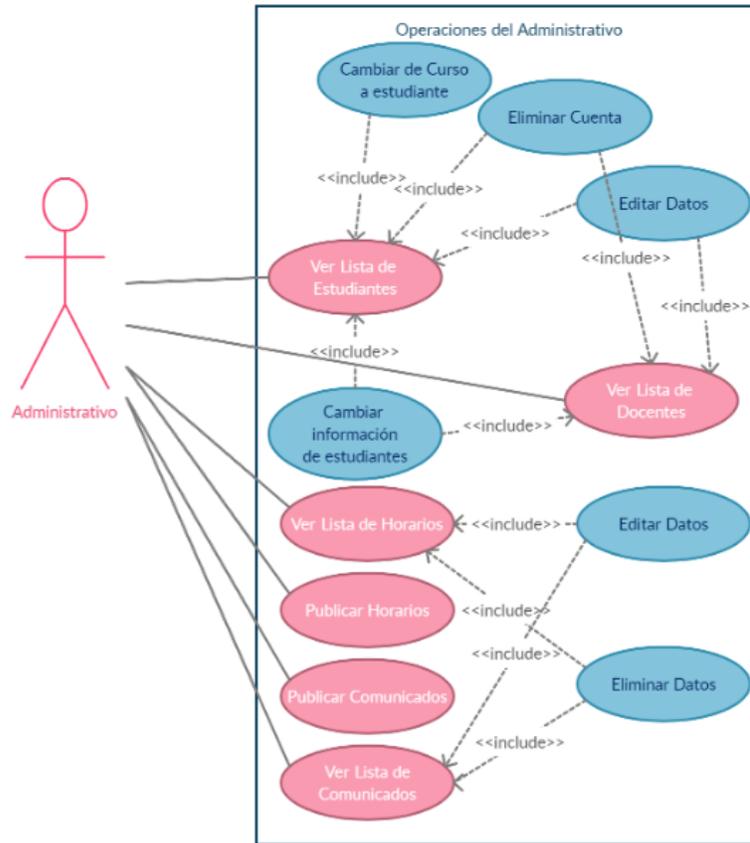


Figura 20: Casos de Uso: Operaciones del Administrativo

Se procede a detallar los casos de uso más importantes de un usuario Administrativo.

Tabla 8: Especificación del caso de uso Cambiar de Curso a estudiante

Nombre	Cambiar de Curso a estudiante
Descripción	Es la operación de un usuario con rol de administrador para cambiar el rol de un estudiante
Actores	Personal Administrativo como director o secretario
Precondiciones	El actor ya ha ingresado al sistema y tiene un rol de administrativo en el sistema.

Postcondiciones	Se ha realizado los cambios de información
Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema con una cuenta valida. 2. Se ingresa a la opción de Lista de Estudiantes 3. El usuario selecciona un curso para acceder a la lista de estudiantes. 4. El sistema muestra la lista de estudiantes con la opción de cambiar el rol de cada estudiante. 5. El actor selecciona los roles que desea cambiar. 6. El sistema guarda los cambios en la base de datos, si se realizaron cambios se muestra un mensaje de confirmación.
Flujo Alternativo	<p>El sistema falla al realizar los cambios en la base de datos</p> <p>En el paso 6 si el sistema falla al enviar los datos se muestra un mensaje de error de la operación.</p>

3.5.2.2 Casos de Uso: Operaciones del Docente

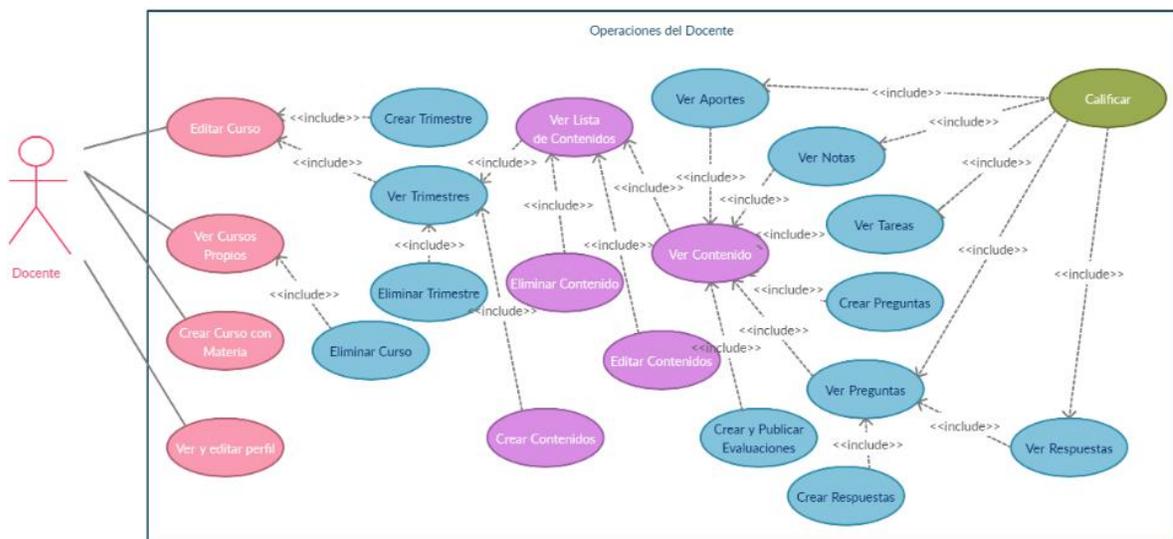


Figura 21: Casos de Uso: Operaciones del Docente

Los casos de uso de operaciones del docente son escalonados y variados por lo que se detallan a continuación los más complejos e importantes para el sistema.

Tabla 9: Especificación del caso de uso Ver Cursos Propios

Nombre	Ver Cursos Propios
Descripción	Es la operación del actor en donde el actor visualiza los cursos pertenecientes a él.
Actores	Docentes de la institución
Precondiciones	El actor ya ha ingresado al sistema y tiene un rol de docente en el sistema.
Postcondiciones	Se ha realizado los cambios de información en la base de datos
Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema con su cuenta. 2. El sistema verifica si el actor tiene una cuenta habilitada y muestra en pantalla el nombre del usuario. 3. El actor selecciona la opción de mis cursos en el menú de opciones.
Flujo Alternativo	<p>El Usuario no tiene ningún curso creado</p> <p>En el paso 4 si el actor no tiene cursos creados no puede acceder a la opción de editar curso.</p>

Tabla 10: Especificación del caso de uso Crear Curso

Nombre	Crear Curso
Descripción	Es la operación del actor en donde realiza la creación de sus cursos.
Actores	Docentes de la institución
Precondiciones	El actor ya ha ingresado al sistema y tiene un rol de docente en el sistema.
Postcondiciones	Se ha realizado los cambios de información en la base de datos
Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema con su cuenta. 2. El sistema verifica si el actor tiene una cuenta habilitada y muestra en pantalla el nombre del usuario.

	<ol style="list-style-type: none"> 3. El actor selecciona la opción de mis cursos en el menú de opciones. 4. El actor selecciona la opción de Crear Curso. 5. El sistema muestra las opciones de creación de curso. 6. El actor escoge las opciones y crea el curso. 7. El sistema guarda las operaciones realizadas y muestra un mensaje de confirmación.
Flujo Alternativo	<p>El sistema falla al crear el curso</p> <p>En el paso 7 si la transacción de creación de curso falla se muestra un mensaje de error.</p>

Tabla 11: Especificación del caso de uso Editar Curso

Nombre	Editar Curso
Descripción	Es la operación del actor en donde realiza la organización de cada curso.
Actores	Docentes de la institución
Precondiciones	El actor ya ha ingresado al sistema y tiene un rol de docente en el sistema.
Postcondiciones	Se ha realizado los cambios de información en la base de datos
Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema con su cuenta. 2. El sistema verifica si el actor tiene una cuenta habilitada y muestra en pantalla el nombre del usuario. 3. El actor selecciona la opción de mis cursos en el menú de opciones. 4. El actor selecciona la opción de editar de la lista de cursos disponibles. 5. Una vez ingresado el sistema le da las opciones de crear trimestres, crear contenidos. 6. El actor empieza a organizar y crear la estructura del curso

	7. El sistema guarda las operaciones realizadas y muestra un mensaje de confirmación.
Flujo Alternativo	<p>El Usuario no tiene ningún curso creado</p> <p>En el paso 4 si el actor no tiene cursos creados no puede acceder a la opción de editar curso.</p> <p>El curso no tiene la organización por trimestres</p> <p>En el paso 5 si el curso aún no está configurado por trimestre el actor no puede crear contenidos para el curso.</p>

Tabla 12: Especificación del caso de uso Calificar

Nombre	Calificar
Descripción	Es la operación de evaluar y asignar una puntuación para las participaciones que tenga el estudiante.
Actores	Docentes de la institución
Precondiciones	El actor ya ha ingresado al sistema y tiene un rol de docente en el sistema.
Postcondiciones	Se ha realizado los cambios de información en la base de datos
Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema con su cuenta. 2. El sistema verifica si el actor tiene una cuenta habilitada y muestra en pantalla el nombre del usuario. 3. El actor selecciona la opción de mis cursos en el menú de opciones. 4. El actor selecciona un contenido. 5. El sistema muestra un formulario de calificación para cada pregunta, respuesta, aporte, nota y tarea entregada por los estudiantes. 6. El actor envía la calificación. 7. El sistema guarda la calificación en la base de datos.

Flujo Alternativo	<p>El contenido ya ha sido calificado</p> <p>En el paso 5 si el contenido ya tiene una calificación asignada el sistema solo mostrará la calificación.</p>
--------------------------	---

Tabla 13: Especificación de Caso de Uso Crear y Publicar Evaluaciones

Nombre	Crear y Publicar Evaluaciones
Descripción	Es la operación de crear una evaluación a asignarlo a un trimestre en una materia y curso para posteriormente los estudiantes puedan acceder a la evaluación
Actores	Docentes de la institución
Precondiciones	El actor ya ha ingresado al sistema y tiene un rol de docente en el sistema.
Postcondiciones	Se ha realizado los cambios de información en la base de datos
Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema con su cuenta. 2. El sistema verifica si el actor tiene una cuenta habilitada y muestra en pantalla el nombre del usuario. 3. El actor selecciona la opción de mis cursos en el menú de opciones. 4. El actor selecciona la opción de crear Evaluaciones. 5. El sistema muestra un formulario con los controles para crear una Evaluación. 6. El actor termina la creación con la opción de Crear Evaluación. 7. El sistema guarda la Evaluación en el sistema y publica la evaluación en el correspondiente Curso Materia y trimestre al que pertenece.

3.5.2.3 Casos de Uso: Operaciones del Estudiante

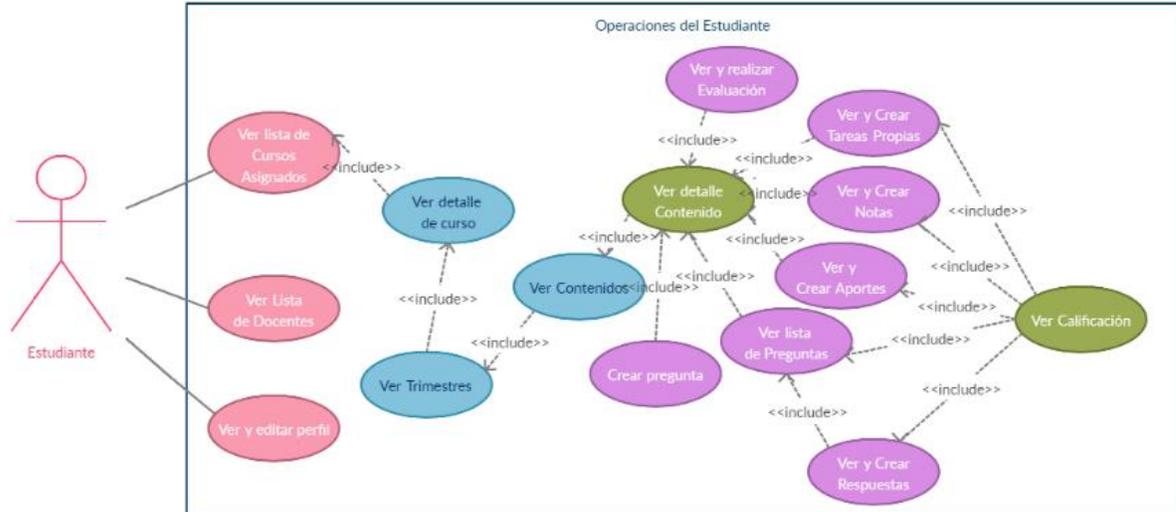


Figura 22: Casos de Uso: Operaciones del Estudiante

Las siguientes tablas detallan los casos de uso más importantes de un usuario Estudiante.

Tabla 14: Especificación del caso de uso Ver lista de cursos asignados

Nombre	Ver lista de cursos asignados
Descripción	Son las operaciones que un usuario con el rol estudiante puede realizar en el sistema
Actores	Estudiantes
Precondiciones	El actor ya ha ingresado al sistema, tiene un rol de estudiante en el sistema y está registrado como estudiante de un curso
Postcondiciones	Se ha realizado los cambios de información
Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema con su cuenta. 2. El sistema verifica si el actor tiene una cuenta habilitada y muestra en pantalla el nombre del usuario. 3. El sistema habilita el menú de opciones para estudiantes. 4. El actor ingresa a la opción de mis cursos.

	<ol style="list-style-type: none"> 5. El sistema muestra los cursos habilitados para el usuario según el nivel al que pertenece. 6. El actor ingresa a un curso de su preferencia. 7. El sistema muestra el contenido del curso como: Contenidos publicados por el docente, enlaces publicados, preguntas del contenido.
Flujo Alternativo	<p>No hay cursos o contenidos disponibles</p> <p>En el paso 5 si el sistema no encuentra cursos disponibles para el usuario, el sistema muestra un texto de información indicando al usuario indicando que no hay cursos o contenidos disponibles.</p>

Tabla 15: Especificación del caso de uso Crear una pregunta

Nombre	Crear una pregunta
Descripción	Son las operaciones que un usuario con el rol estudiante puede realizar en cada contenido.
Actores	Estudiantes
Precondiciones	El actor ya ha ingresado al sistema, tiene un rol de estudiante en el sistema y está registrado como estudiante de un curso.
Postcondiciones	Se ha realizado los cambios de información
Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema con su cuenta. 2. El sistema verifica si el actor tiene una cuenta habilitada y muestra en pantalla el nombre del usuario. 3. El sistema habilita el menú de opciones para estudiantes. 4. El actor ingresa a la opción de mis cursos. 5. El sistema muestra los cursos habilitados para el usuario según el nivel al que pertenece. 6. El actor ingresa a un curso de su preferencia.

	<p>7. El sistema muestra el contenido del curso como: Contenidos publicados por el docente, enlaces publicados, preguntas del contenido.</p> <p>8. El actor selecciona un contenido para visualizarlo.</p> <p>9. El sistema muestra las opciones de crear una pregunta, crear un aporte, crear una nota, crear una tarea.</p> <p>10. El actor escribe y crea una pregunta.</p> <p>11. El sistema la muestra un mensaje de confirmación al actor.</p> <p>12. Una vez confirmada la operación el sistema guarda en la base de datos la pregunta.</p>
Flujo Alternativo	<p>El contenido ha sido bloqueado por el docente</p> <p>En el paso 9 si el contenido está bloqueado por el docente, el actor no podrá realizar ninguna pregunta.</p> <p>El usuario a alcanzado el límite máximo de preguntas por trimestre.</p> <p>En el paso 10 si el actor ha realizado 50 preguntas en el trimestre no podrá realizar más preguntas y se le mostrará un mensaje de error.</p>

Tabla 16: Especificación del caso de uso Ver Calificación

Nombre	Ver Calificación
Descripción	La visualización de calificaciones que puede hacer el estudiante.
Actores	Estudiantes
Precondiciones	El actor ya ha ingresado al sistema, tiene un rol de estudiante en el sistema y está registrado como estudiante de un curso.
Postcondiciones	Se ha realizado los cambios de información
Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema con su cuenta. 2. El sistema verifica si el actor tiene una cuenta habilitada y muestra en pantalla el nombre del usuario.

	<ol style="list-style-type: none"> 3. El sistema habilita el menú de opciones para estudiantes. 4. El actor ingresa a la opción de mis cursos. 5. El sistema muestra los cursos habilitados para el usuario según el nivel al que pertenece. 6. El actor ingresa a un curso de su preferencia. 7. El sistema muestra el contenido del curso como: Contenidos publicados por el docente, enlaces publicados, preguntas del contenido. 8. El actor selecciona un contenido para visualizarlo. 9. El sistema muestra las opciones de ver preguntas, ver respuestas, ver aportes, ver contenidos y ver tareas. 10. El actor selecciona una opción. 11. El sistema la muestra todas las calificaciones dadas por el docente.
Flujo Alternativo	<p>No existe una calificación</p> <p>En el paso 11 si no se tiene una calificación el sistema no muestra ninguna información.</p>

Tabla 17: Ver y Realizar evaluación

Nombre	Ver y Realizar evaluación
Descripción	Es la operación del estudiante para acceder a las evaluaciones que publican los docentes
Actores	Estudiantes
Precondiciones	El actor ya ha ingresado al sistema, tiene un rol de estudiante en el sistema y está registrado como estudiante de un curso.
Postcondiciones	Se ha realizado los cambios de información
Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema con su cuenta.

	<ol style="list-style-type: none"> 2. El sistema verifica si el actor tiene una cuenta habilitada y muestra en pantalla el nombre del usuario. 3. El sistema habilita el menú de opciones para estudiantes. 4. El actor ingresa a la opción de mis cursos. 5. El sistema muestra los cursos habilitados para el usuario según el nivel al que pertenece. 6. El actor ingresa a un curso de su preferencia. 7. El sistema muestra el contenido del curso como: Contenidos publicados por el docente, enlaces publicados, preguntas del contenido y evaluaciones publicadas. 8. El actor selecciona una evaluación disponible. 9. El sistema muestra la información inicial de la evaluación y la opción de iniciar con la Evaluación. 10. El actor selecciona una la opción de Iniciar Evaluación. 11. El sistema muestra uno a uno las preguntas de opción múltiple de la Evaluación. 12. El sistema Muestra los resultados que obtuvo el estudiante. 13. El actor escoge entre las opciones de volver a intentar y enviar los resultados obtenidos. 14. El sistema guarda en la base de datos los resultados que obtuvo el estudiante.
Flujo Alternativo	<p>El estudiante presiona la opción de volver a intentar</p> <p>En el paso 13 si el estudiante tiene dos intentos ya realizados en la evaluación esta opción es deshabilitada.</p>

3.5.2.4 Diagrama de Clases: Módulo de Servicios

Se muestra en el diagrama de clases todas las colecciones necesarias para guardar los datos importantes en cada operación que realicen los usuarios.

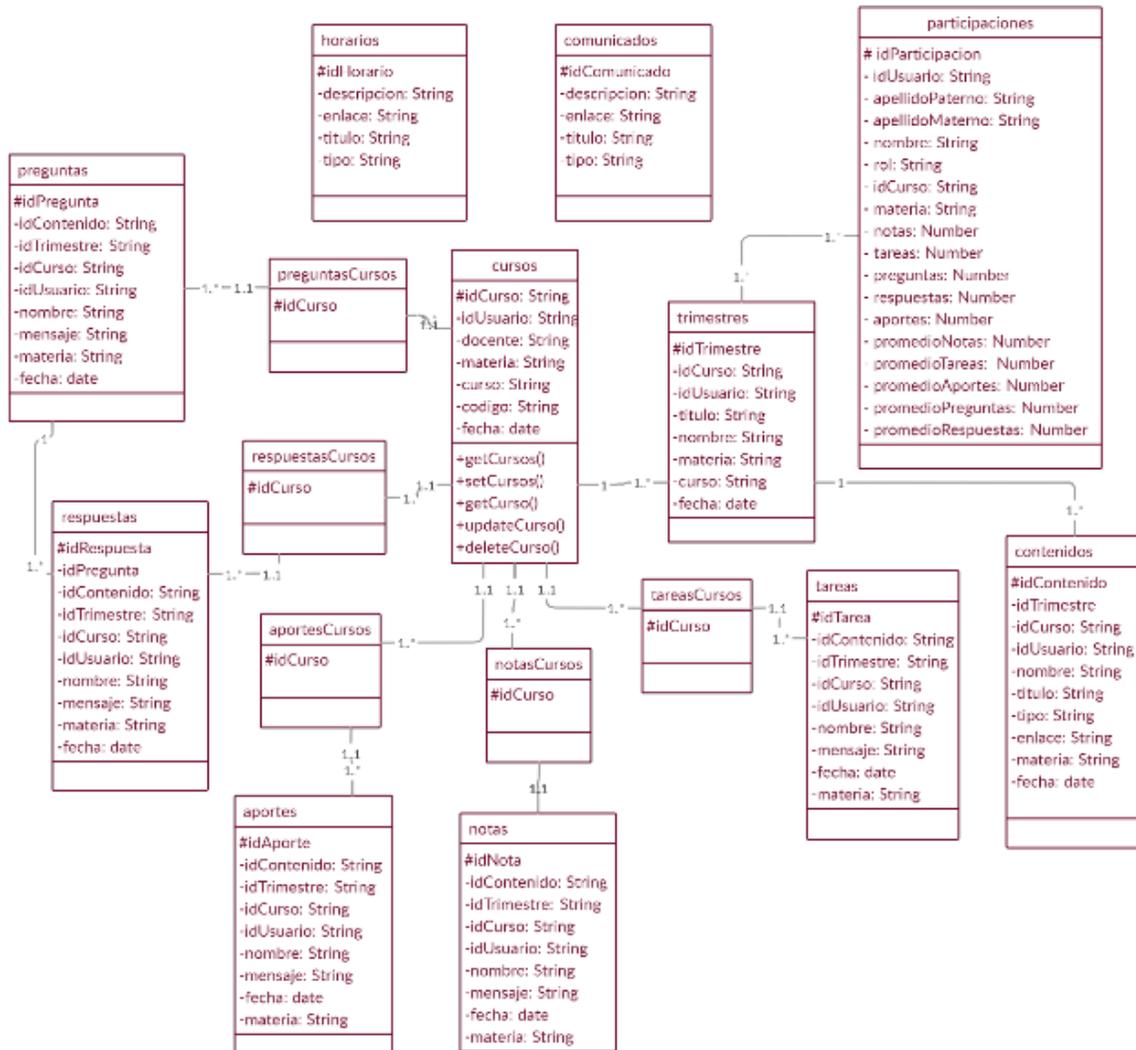


Figura 23: Casos de Uso: Operaciones del Docente

3.5.2.5 Tablero Kanban

Al ser el Módulo de Servicios parte central del sistema se especifica y divide en tareas. Esta división de tareas se muestra en la siguiente figura.

Opción para cambiar de curso a estudiante	Publicar Comunicados
Eliminar Cuenta	Ver lista de Comunicados
Editar Datos	Editar datos de Horario y Comunicados
Ver lista de estudiantes	Eliminar datos de Horarios y comunicados
Ver lista de docentes	Crear curso con materia
Cambiar información de estudiantes	Ver editar perfil
Publicar Horarios	Eliminar contenido
Ver cursos propios	Ver contenido
Editar curso	Editar contenidos
Crear Trimestre	Ver aportes
Ver Trimestres	ver Notas
Eliminar Trimestres	Ver Tareas
Eliminar Curso	Crear Preguntas
Ver lista de contenidos	Ver Preguntas

Crear Respuestas
Ver respuestas
Calificar Aportes, notas, tareas, preguntas, respuestas,

Figura 24: Tablero Kanban Tareas para el módulo de servicios

3.5.3 Fase de Implementación

En la fase de implementación se trabajará considerando el WIP de 2 tareas y utilizaremos el diagrama de estado para representar el funcionamiento del sistema.

3.5.3.1 Diagrama de estado: Operaciones del Administrativo

El diagrama de estado muestra las páginas y su comportamiento accediendo al sistema. El usuario al iniciar sesión se encontrará con la página Home y la Barra de Navegación en donde se le mostrarán las operaciones que puede realizar como administrador.

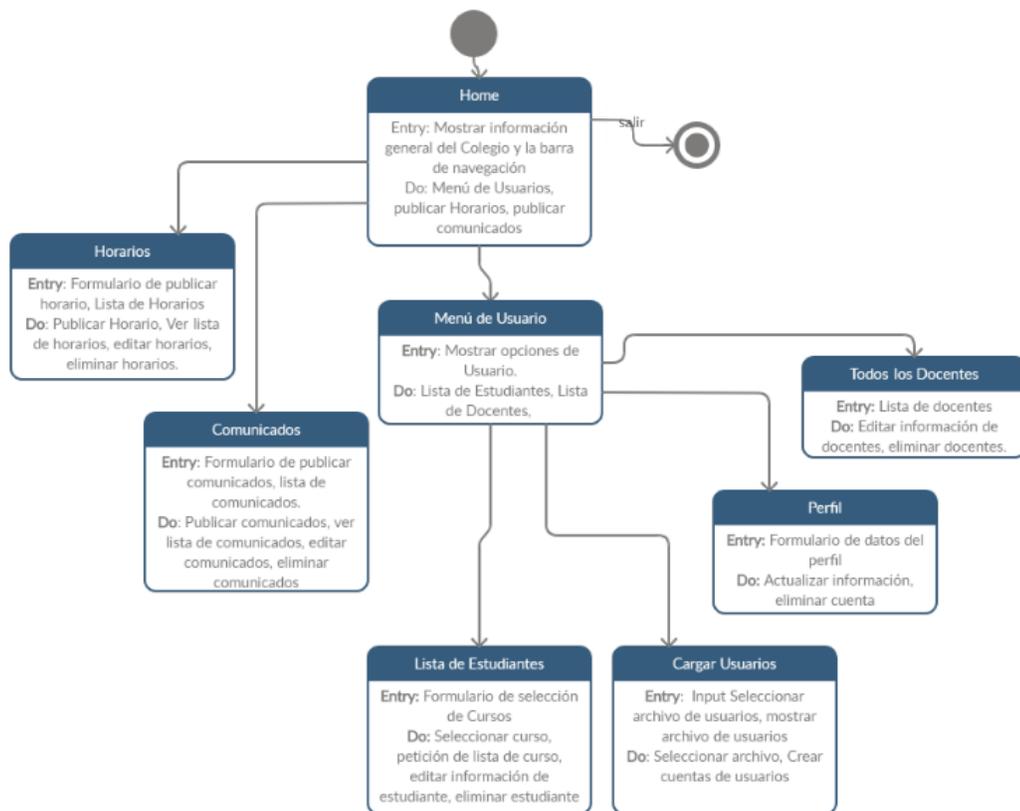


Figura 25: Diagrama de estado: Operaciones de Administrativo

3.5.3.2 Diagrama de estado: Operaciones del Docente

En el diagrama se muestran las operaciones que puede realizar el docente en la aplicación web.

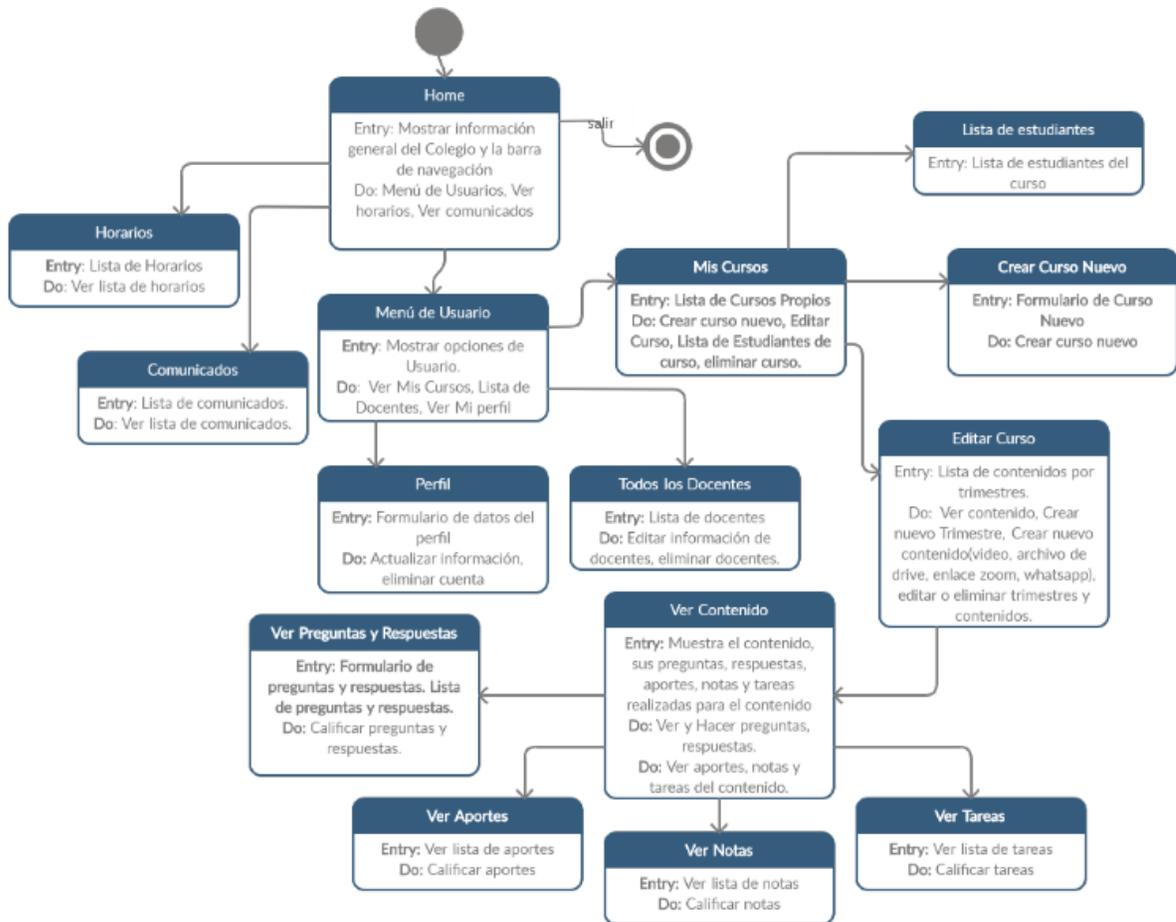


Figura 26: Diagrama de estado: Operaciones de Docente

3.5.3.2 Diagrama de estado: Operaciones del Estudiante

En el diagrama se muestran las operaciones que puede realizar un usuario estudiante en la aplicación web.

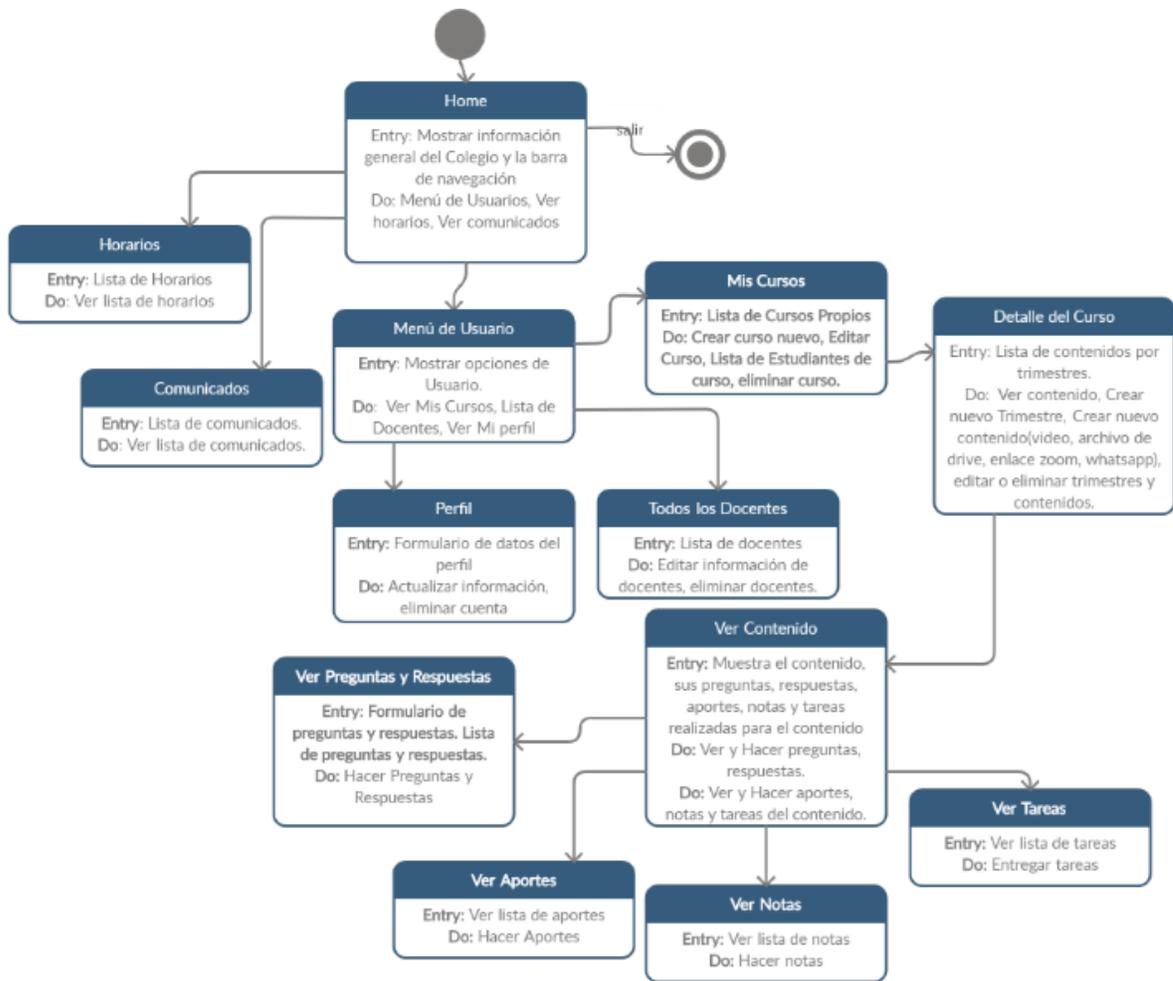


Figura 27: Diagrama de estado: Operaciones de Estudiante

3.5.4 Fase de Validación

Para el módulo de servicios la revisión de código y la verificación de pruebas se realiza en conjunto con el módulo de administración de usuarios para verificar una correcta integración entre los dos módulos.

Tabla 18: Validación de Módulo de Servicios

Validación de Servicios	
Validaciones	Estado(SI/NO)
Revisión del Código	SI
Verificar pruebas unitarias	SI

Verificación de la documentación	SI
Integración del Módulo al repositorio	SI
Despliegue de código a servidor	SI

3.5.4.1 Pruebas unitarias

En esta fase se realizan las pruebas unitarias necesaria para el módulo de servicios del sistema.

Tabla 19: Pruebas Unitarias Módulo de Servicios

Pruebas	Resultado
El componente MisCursosDocente se muestra correctamente	Cumple
El componente MisCursosEstudiante se muestra correctamente	Cumple
El componente de PreguntasList se muestra correctamente	Cumple
El componente de RespuestasList se muestra correctamente	Cumple
El componente ClassroomForm se muestra correctamente	Cumple
El componente GeneralList se muestra correctamente	Cumple

En la siguiente figura se muestran los resultados obtenidos en la consola que demuestran el correcto funcionamiento de los componentes de forma individual.

```

FBIL src/_test_/components/Servicios.test.js (9.066s)
Tests de modulo de servicios
  × Render del componente MisCursosDocente (31ms)
  × Render del componente MisCursosEstudiante (16ms)

• Tests de modulo de servicios › Render del componente MisCursosDocente

TypeError: Cannot read property 'rol' of undefined

28 |         return unsubscribe;
29 |     }, [])
> 30 |     if (usuarioLogeado.rol !== 'docente') {
    |                         ^
31 |         return <h1 className="error">Ups! No tienes acceso a estos datos.</h1>
32 |     } else {
33 |         return (

```

Figura 28: Resultado de pruebas unitarias módulo de servicios

Se observa que se tiene un error de una propiedad no definida el cual se presenta porque la propiedad es obtenida del resultado de una petición a la base de datos por lo que se debe de corregir la propiedad del componente.

3.6 MÓDULO DE REPORTES

3.6.1 Fase de Pedidos

Detallamos la lista de pedidos sobre los reportes del sistema:

- Personal Administrativo tendrá la opción de imprimir reportes sobre: lista general de estudiantes, lista general de docentes, reportes trimestrales y reportes por curso.
- Los docentes tendrán la opción de generar reportes de estudiantes de los cursos, reportes de calificaciones de estudiantes, reportes de estadísticas de los cursos y reporte de filiación de los cursos.
- Los estudiantes tendrán la opción de tener la lista de cursos inscritos, lista de puntajes obtenidos y su detalle.

3.6.2 Fase de Especificación

3.6.2.1 Casos de Uso: Reportes para Usuarios

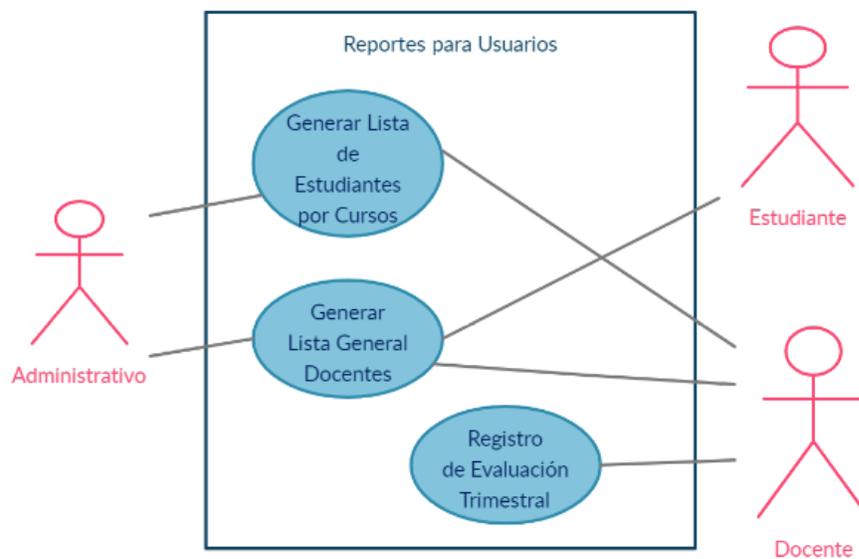


Figura 29: Caso de Uso: Reportes para el Personal Administrativo

Tabla 20: Especificación del caso de uso Generar Lista de Estudiantes

Nombre	Generar Lista de Estudiantes y Docentes
Descripción	Es la operación que el actor realiza para obtener la información de los estudiantes en un archivo en formato pdf.
Actores	Administrativos y Docentes
Precondiciones	El actor ya ha ingresado al sistema, tiene un rol de administrador o docente en el sistema.
Postcondiciones	
Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema con su cuenta. 2. El sistema verifica si el actor tiene una cuenta habilitada y muestra en pantalla el nombre del usuario. 3. El sistema habilita el menú de opciones 4. El actor ingresa a la opción de Lista de Estudiantes o Lista de Docentes. 5. El sistema muestra la lista de estudiantes o docentes con la opción de generar un archivo en pdf. 6. El actor selecciona la opción de generar archivo. 7. El sistema guarda la información de la base de datos en un archivo pdf.

Tabla 21: Especificación del caso de uso Registro de Evaluación Trimestral

Nombre	Registro de Evaluación Trimestral
Descripción	Es la operación que el actor realiza para obtener la información de los estudiantes en un archivo en formato pdf.
Actores	Docentes
Precondiciones	El actor ya ha ingresado al sistema, tiene un rol de docente en el sistema.
Postcondiciones	

Flujo Normal	<ol style="list-style-type: none"> 1. El Actor ingresa al sistema con su cuenta. 2. El sistema verifica si el actor tiene una cuenta habilitada y muestra en pantalla el nombre del usuario. 3. El sistema habilita el menú de opciones 4. El actor ingresa a la opción cursos. 5. El actor selecciona un curso. 6. El sistema muestra las opciones de generar Registro de evaluación trimestral 7. El actor seleccionar la opción de generar archivo pdf. 8. El sistema guarda la información de la base de datos en un archivo pdf.
---------------------	---

3.6.2.2 Tablero Kanban

En el módulo de reportes las tareas a realizar son consultas a la base de datos. Con la estructura de una base de datos noSql se utiliza consultas con pocas condicionales y son más fáciles de estructurar en los reportes.

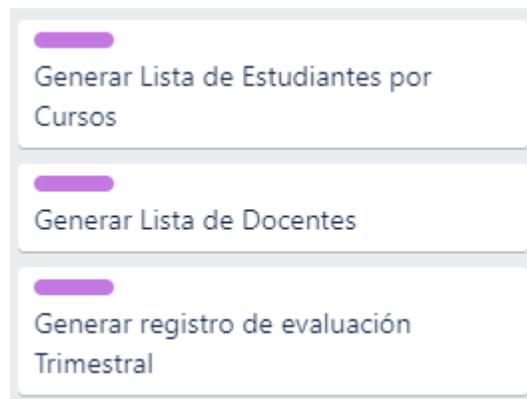


Figura 30: Tablero: Módulo de Reportes

3.6.3 Fase de Implementación

3.6.3.1 Diagrama de estado: Generar Reportes

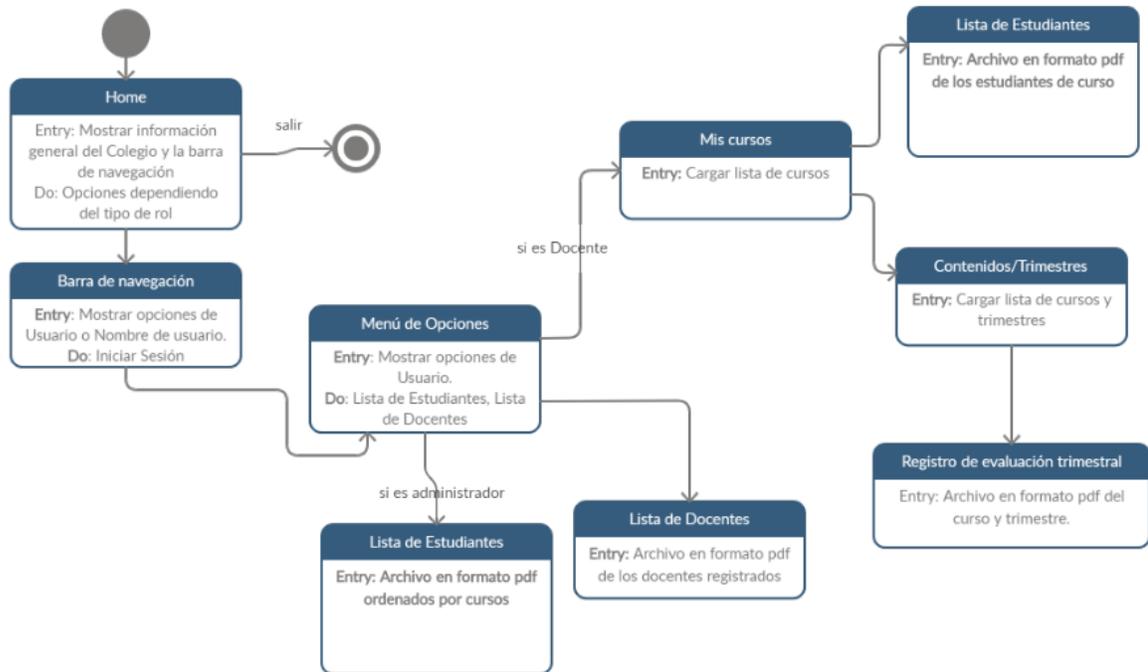


Figura 31: Diagrama de estado: Generar Reportes

3.6.4 Fase de Validación

En la fase validación para el módulo de reportes se realiza una integración previa con el módulo de servicios para verificar el correcto funcionamiento del sistema. Esto debido a la dependencia del módulo de reportes con el módulo de servicios.

Tabla 22: Validación de Módulo de Reportes

Validación de Reportes	
Validaciones	Estado(SI/NO)
Revisión del Código	SI
Verificar pruebas unitarias	SI
Verificación de la documentación	SI
Integración del Módulo al repositorio	SI

Despliegue de código a servidor	SI
---------------------------------	----

3.6.4.1 Pruebas unitarias

En esta fase se realizan las pruebas unitarias necesaria para el módulo de reportes del sistema.

Tabla 23: Pruebas Unitarias Módulo de Reportes

Pruebas	Resultado
El componente GeneralList se muestra correctamente	Cumple
El componente EstudiantesList se muestra correctamente	Cumple

En la siguiente figura se muestran los resultados obtenidos en la consola que demuestran el correcto funcionamiento de los componentes de forma individual.

```

FAIL src/__test__/components/Reportes.test.js (8.652s)
Tests de modulo de servicios
  ✓ Render del componente GeneralList (31ms)
  ✗ Render del componente EstudiantesList (302ms)

  • Tests de modulo de servicios › Render del componente EstudiantesList

    TypeError: Cannot read property 'params' of undefined

    53 |     }
    54 |     const getCurso = () => {
  > 55 |       db.collection('trimestres').doc(props.match.params.idTrimestre).get().then(doc => {
      |                                     ^
  
```

Figura 32: Resultado de pruebas unitarias módulo de reportes

Se observa que se tiene un error de una propiedad params del componente el cual se obtiene de la ruta generada para el usuario por lo que se debe de simular esta propiedad para futuras pruebas.

También se observa que el componente GeneralList no tiene ningún problema en la prueba.

CAPITULO IV

4. CALIDAD Y SEGURIDAD

En este capítulo se describe la calidad del sistema basado en los parámetros de la norma ISO 9126, esto para establecer la base para la seguridad, costos y beneficios que tiene el sistema.

4.1 CALIDAD ISO 9126

Se realizan un conjunto de medidas conocidas como métricas, las cuales proporcionan una referencia de la calidad en un producto de software (Pressman, 2010).

Medir la calidad de un software es en cierta manera una tarea complicada en el desarrollo de un sistema. Pero gracias a esta necesidad se fueron creando diferentes formas de medición de las mismas. (Pressman, 2010).

Para el presente proyecto se utiliza la metodología **WebQem** para medir la calidad del sistema tomando en cuenta las características del sistema.

Así mismo para obtener los resultados y métricas se utiliza encuestas realizadas a los actores principales como ser el Personal docente, administrativo y los estudiantes en donde en el ANEXO H se puede acceder a los enlaces de dichas encuestas y resultados obtenidos.

4.1.1 Funcionalidad

Para calcular la funcionalidad utilizamos las ecuaciones 1 y 2 y representamos los resultados en la siguiente tabla.

Tabla 24: Prueba de Funcionalidad del Sistema

Nro.	Encuesta de Funcionalidad	Regular (60%)	Bueno (80%)	Muy Bueno (100%)
1	¿Los datos de registro son suficientes o se requieren datos adicionales?		X	

2	¿Se mejoró la gestión de información?			X
3	¿Los reportes tienen la información necesaria?		X	
4	¿Las evaluaciones son realizadas correctamente y son entendibles?			X
Promedio				90%

La funcionalidad es del 90%.

Considerando que se tiene el 90% de funcionalidad significa que a 1 de 10 usuarios tuvo inconvenientes, el resultado que se tiene se encuentra en Muy Bueno pero con un margen para mejorar.

4.1.2 Usabilidad

Se realiza una encuesta para determinar la usabilidad del sistema.

Tabla 25: Encuesta de usabilidad del sistema

Nro.	Pregunta	SI	NO	% DE SI
1	¿El aprendizaje del sistema es sencillo?	10	5	67
2	¿Las pantallas de administración son agradables?	12	3	80
3	¿Las pantallas son comprensibles?	12	3	80
4	¿El sistema responde rápido a las solicitudes?	14	1	93
5	¿El sistema facilita su trabajo?	10	5	67
6	¿El sistema reduce su tiempo de trabajo?	13	2	87
7	¿La navegación por las opciones es sencilla?	12	3	80
8	¿Las operaciones que realiza no son complicadas?	13	2	87
9	¿El sistema proporciona la información requerida?	12	3	80

10	¿El sistema no presento errores?	12	3	80
Promedio				80%

Se tiene un promedio de 80% de confiabilidad, esto implica decir que 8 de cada 10 usuarios confían en el sistema y 2 usuarios tiene dudas, según los parámetros establecidos el resultado se encuentra en el parámetro de Bueno.

4.1.3 Mantenibilidad

Tomando en cuenta los datos necesarios para la ecuación 5 se muestra la siguiente tabla:

Tabla 26: Mantenibilidad del sistema

Información	Valor
Mt	3
Fa	0
Fb	0
Fc	0

Se calcula el índice de madurez del software, usando los valores que se tiene. Pero se debe de considerar que no se contaba con un sistema anterior por lo que la ecuación tendría un impacto menor en sus resultados.

$$IMS = [3-(0+0+0)]/3$$

$$IMS = 3/3=1=100\%$$

La ecuación refleja el máximo porcentaje de mantenibilidad. Lo que no reflejaría necesariamente la realidad al no contarse con un sistema anterior de comparación.

4.1.4 Confiabilidad

Para determinar la confiabilidad de un sistema, se toma en cuenta las fallas que puedan suceder en el sistema. Entre las fallas más comunes que se tienen son por

el diseño e implementación. Con lo cual se calcula el tiempo medio entre fallos con la siguiente fórmula:

$$TMEF = TMDF + TMDR$$

Donde:

TMDF = Tiempo medio de fallo.

TMDR = Tiempo medio de reparación

Se estima que un fallo pueda ocurrir entre un intervalo de 14 días y la reparación de cada fallo pueda tomar en promedio 1 hora después de haber entregado una nueva funcionalidad del sistema.

Por lo que se tiene:

$$TMEF = (14 * 8) + 2$$

$$TMEF = 114 \text{ horas}$$

Tomando en cuenta la disponibilidad como un indicador, en base al siguiente calculo:

$$\text{Disponibilidad} = TMDF/TMEF * 100$$

$$\text{Disponibilidad} = \frac{112}{114} * 100 = 98.24\%$$

Con lo que tenemos como resultado que el sistema tiene un 98.24% de confiabilidad.

4.1.5 Portabilidad

Para determinar el grado de portabilidad que el software tiene para ser llevado de un entorno a otro, se considera la facilidad de instalación, ajuste y adaptación al cambio.

Se usa la siguiente fórmula:

$$GP = 1 - \frac{ET}{ER}$$

Donde:

GP = Grado de portabilidad

ET = Recursos necesarios para llevar el sistema a otro entorno.

ER = Recursos necesarios para crear el sistema en el entorno residente.

Los recursos necesarios para llevar el sistema a otro entorno son: el hosting, la base de datos, el dominio, conexión al repositorio, conexión a internet.

Los recursos para crear el sistema son: Un editor de texto Visual Studio Code, framework React js, html5, css3, node js, maquetador adobe xd.

Entonces tenemos los valores:

$$ET = 5$$

$$ER = 4$$

$$GP = 1 - \frac{5}{7}$$

$$GP = 0.28$$

Con el resultado obtenido se tiene que el grado de portabilidad es de 0.28, con lo que la portabilidad del sistema es más rentable que el re desarrollo.

4.1.4 Calidad Global

Se toma en cuenta las características analizadas y cuantificadas anteriormente, esto para entender en que porcentaje de calidad se está manejando el proyecto y analizar el resultado para así tomar decisiones futuras.

Tabla 27: Calidad Global

Características de calidad	Porcentaje
Funcionalidad	90
Usabilidad	80
Mantenibilidad	100
Confiabilidad	98

Portabilidad	28
Total	79

El promedio global de la calidad que se obtuvo es de 79% esto refleja que el sistema es efectivo y el producto es aceptable para el momento de la medición de la calidad. Se debe de considerar que este tipo de pruebas pueden hacerse de forma periódica en próximas actualizaciones del sistema.

4.2 SEGURIDAD

Para que un sistema sea considerado seguro, deben considerarse las posibles amenazas y, por lo tanto, conocer y prever las acciones de los intrusos.

4.2.1 Seguridad Física

Las amenazas de tipo accidentales o naturales como ser incendios e inundaciones el sistema tiene desde un principio con un almacenamiento del proyecto en repositorios remotos y la base de datos resguardada en un servidor de Firebase.

4.2.2 Seguridad Lógica

Para la seguridad lógica del sistema se toma en cuenta los tres objetivos principales:

4.2.2.1 Integridad

Para cubrir la integridad de los datos almacenados en el sistema se hizo la creación de roles específicos para ciertos usuarios, lo que garantiza que solo usuarios con estos roles hagan modificaciones en el sistema.

4.2.2.2 Confidencialidad

Para poder cumplir con este objetivo se realizó la configuración de reglas de seguridad en el acceso a la lectura y escritura en la base de datos. Con estas reglas de seguridad establecidas se garantiza que solo los usuarios puedan acceder y modificar los datos generados por ellos mismos.

4.2.2.3 Disponibilidad

Para asegurar la disponibilidad de datos, se utiliza el servicio de autenticación *Firebase* de sesión permitiendo el acceso a la información autorizada sin

restricciones de ningún tipo, pero si el sistema es modificado por un usuario administrador el sistema niega las sesiones entrantes de los usuarios.

Se aclara que el manejo de usuarios y contraseñas es manejado por el mismo servicio de autenticación de *Firebase*.

CAPITULO V

5. ANALISIS DE COSTOS

Para el análisis de costos se tomará en cuenta el costo de desarrollo de la aplicación y no así del costo de equipos de computación ya que en la institución y los usuarios usarán equipos y conexión propia.

5.1 COSTO DEL SOFTWARE

5.1.1 Punto de Función

Como primera acción se halla los puntos de función no ajustados “PFNA” para esto se inicia con la identificación de 5 características que se detalla a continuación:

5.1.1.1 Número de entradas de usuario

Es la información en datos que se ingresan al sistema para realizar las distintas operaciones.

Tabla 28: Entradas de usuario

Nro.	Entradas
1	Registro de datos de usuario
2	Registro de Rol de usuario
3	Registro de Curso de Usuario
4	Registro de Curso
5	Registro de Contenido
6	Registro de Trimestre
7	Registro de Comunicados y Horarios
8	Registro de Preguntas, Respuestas, Aportes, Notas y Tareas
9	Registro de calificaciones
10	Registro de Participaciones

5.1.1.2 Número de salidas de usuario

Se presentan la información elaborada por el sistema que es gestionada por los usuarios.

Tabla 29: Salidas de usuario

Nro.	Salidas
1	Información de usuario
2	Lista de curso creados
3	Lista de cursos disponibles
4	Lista de contenidos
5	Lista de trimestres
6	Lista de Preguntas, Respuestas, Aportes, Notas y Tareas
7	Lista de docentes
8	Lista de Estudiantes por curso
9	Lista de calificaciones trimestrales
10	Lista de Comunicados
11	Lista de Horarios
12	Lista de estudiantes de curso

5.1.1.3 Número de peticiones de usuario

Tabla 30: Peticiones de usuario

Nro.	Peticiones
1	Autenticación
2	Lista generada de estudiantes de Curso
3	Lista generada de Docentes
4	Lista generada de Calificaciones por trimestre

5.1.1.4 Número de archivos

Es un grupo lógico de datos en el caso de este proyecto son todas las colecciones de la base de datos.

Tabla 31: Número de archivos

Nro.	Archivos
1	Claves
2	FirestoreUsuarios
3	Users
4	Comunicados
5	Horarios
6	Cursos
7	Trimestres
8	Contenidos
9	Tareas
10	Preguntas
11	Respuestas
12	Notas
13	Aportes
14	PreguntasCursos
15	RespuestasCursos
16	NotasCursos
17	AportesCursos
18	TareasCursos
19	Calificaciones

5.1.1.4 Numero de interfaces externas

Existe solo la interfaz de internet a través de un navegador.

Como siguiente paso se agrupan todas las entradas, salidas, peticiones, archivos e interfaces externas junto al factor de ponderación.

Tabla 32: Número de interfaces externas

Parámetros de medición	Cuentas	Factor de Ponderación			Totales
N. de entradas de usuario	10		4		40
N. de salidas de usuario	12		5		60
N. de peticiones de usuario	4		4		16
N. de archivos	20		10		200
N. de interfaces externas	1		7		7
Total en cuenta					323

En la siguiente tabla se muestran los factores de complejidad en base a las respuestas de las siguientes preguntas evaluadas entre 0 y 5.

Tabla 33: Factor de ajuste de complejidad

Factores de complejidad	Sin infidencia	Incidental	Moderado	Medio	Significativo	Esencial	Fi
	0	1	2	3	4	5	

¿Requiere el sistema copias de seguridad y de recuperación fiables?				X			3
¿Se requiere comunicación de datos?					X		4
¿Existen funciones de procesamiento distribuido?			X				2
¿Es crítico el rendimiento?				X			3
¿Se ejecutará el sistema con un entorno operativo existente y fuertemente utilizado?			X				2
¿Requiere el sistema entrada de datos interactiva?						X	5
Facilidad operativa				X			3
¿Se actualiza los archivos maestros de forma interactiva?					X		4
¿Son complejas las entradas, las salidas, los archivos o las peticiones?				X			3
¿Procesamiento interno complejo			X				2
¿Diseño de código reutilizable?					X		4
¿Facilidad de instalación?					X		4
¿Soporta múltiples instalaciones en diferentes sitios?					X		4
Facilidad de cambios					X		4
Factor de ajuste de complejidad							47

Calculamos el factor de ajuste

$$Factor\ de\ ajuste = 0.65 + 0.01 * \sum fi$$

$$Factor\ de\ ajuste = 0.65 + 0.01 * 47$$

$$Factor\ de\ ajuste = 1.12$$

Calculamos el punto de función

$$\text{Punto función} = \text{cuenta total} * \text{factor de ajuste}$$

$$\text{Punto función} = 323 * 1.12$$

$$\text{Punto función} = 361.76$$

5.1.2 Conversión de PF a LDC (líneas de código)

Tomando en cuenta el lenguaje de implementación se considera las líneas de código por punto de función.

Tabla 34: Tabla de conversión factor LDC

Lenguaje	LDC/PF
ASP	56
Assembler	209
C	148
C++	59
C#	58
FoxPro	36
J2EE	57
Java	55
JavaScript	34
React	20
JSP	59
.NET	60

La ecuación para el cálculo de LDC es:

$$LDC = PF * \text{Factor LDC/PF}$$

Después se convierte el punto de función a miles de líneas de código.

$$LDC = 361.76 * 20$$

$$LDC = 7235.2$$

$$KLDC = 7235.2/1000 = 7.2$$

5.1.3 Estimaciones de esfuerzo nominal

El esfuerzo nominal está dado por:

$$PM_{nominal} = A * (KLDC)^B$$

Donde:

$$B = 0.91 + 0.01 * \sum W_j$$

B=Factor exponencial de escala, está basado en factores de escala que influyen exponencialmente en la productividad.

W_j= Factores de escala

PM_{nominal}= Esfuerzo nominal de proyecto

KLDC= Tamaño del software a desarrollar expresado en miles de líneas de código fuente.

A= Constante que captura los efectos lineales sobre el esfuerzo de acuerdo a la variación del tamaño (A=2.94).

En la siguiente tabla se describe el Factor de Escala W_j.

Donde:

PREC = Experiencia en la aplicación del mismo tipo

FLEX = Grado de sujeción del desarrollo a tiempo y requisitos

RESL= Identificación de riesgos en la aplicación

TEAM = Nivel de Integración del equipo de desarrollo

PMAT = Experiencia en el modelo de desarrollo

Tabla 35: Factores de escala Wj

Factores de escala	Muy Bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
PREC	6.2	4.96	3.72	2.48	1.24	0
FLEX	5.07	4.05	3.04	2.03	1.01	0
RESL	7.07	5.65	4.24	2.83	1.41	0
TEAM	5.48	4.38	3.29	2.19	1.1	0
PMAT	7.8	6.29	4.68	3.12	1.56	0

Mediante la ecuación descrita se halla el esfuerzo nominal del proyecto.

$$B = 0.91 + 0.01 * (1.24 + 1.01 + 0 + 1.1 + 0)$$

$$B = 0.94$$

Reemplazando

$$PM_{nominal} = A * (KLDC)^B$$

$$PM_{nominal} = 2.94 * (7.2)^{0.94}$$

$$PM_{nominal} = 18 \text{ personas/mes}$$

El resultado indica que se necesitarían 31 personas trabajando durante un mes, pero es necesario saber el Esfuerzo estimado antes de tomar el resultado.

La ecuación es:

$$PM_{estimado} = PM_{nominal} * \prod_{t=1}^{17} EM_j$$

Donde:

PM_{estimado} = Esfuerzo estimado del proyecto que está basado en los multiplicadores de esfuerzo para su ejecución.

EM_j = Multiplicadores de esfuerzo

PM_{nominal} = Esfuerzo nominal del proyecto

Multiplicadores de esfuerzo

Esto servirán para hallar el esfuerzo inicial y se clasificarán en cuatro grupos que son producto, plataforma, personal y proyecto.

Producto

Son los factores del producto considerados en la variación del esfuerzo requerido causadas por el producto en sí.

Tabla 36: Multiplicadores de esfuerzo

Nro.	Factor	Característica
1	RELY	Fiabilidad requerida del software
2	DATA	Tamaño de la base de datos
3	CPLX	Complejidad del producto
4	RUSE	Reutilización requerida
5	DOCU	Documentación que se desarrollo

Plataforma

Son las restricciones impuestas por el software y hardware.

Tabla 37: Multiplicadores de esfuerzo de plataforma

Nro.	Factor	Característica
6	TIME	Restricciones en el tiempo de ejecución
7	STORE	Restricciones en el almacén principal
8	PVOL	Volatilidad de la plataforma

Personal

Son los factores que toma en cuenta la experiencia y capacidad del personal de desarrollo del proyecto.

Tabla 38: Multiplicadores de esfuerzo de Personal

Nro.	Factor	Característica
9	ACAP	Aptitud de analistas
10	PCAP	Aptitud de programadores
11	AEXP	Experiencia en desarrollo de aplicaciones similares
12	PCON	Continuidad del personal
13	PEXP	Experiencia en la plataforma de desarrollo
14	LEXP	Experiencia en el lenguaje y herramientas

Proyecto

Son las características particulares del proyecto desarrollado.

Tabla 39: Multiplicadores de esfuerzo del proyecto

Nro.	Factor	Característica
15	TOOL	Utilización de herramientas de software
16	SITE	Desarrollo de múltiples localizaciones
17	SCED	Tiempo necesario para el desarrollo

En la siguiente tabla se muestra los multiplicadores de esfuerzo seleccionados.

Tabla 40: Multiplicadores para el costo de proyecto

Conductores de costos	Nro	Multiplicadores de esfuerzo	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Producto	1	RELY	0.82	0.92	1	1.1	1.26	-
	2	DATA	-	0.9	1	1.14	1.28	-

	3	CPLX	0.73	0.87	1	1.17	1.34	-
	4	RUSE	-	0.95	1	1.07	1.15	-
	5	DOCU	0.81	0.91	1	1.11	1.23	-
Plataforma	6	TIME	-	-	1	1.11	1.29	-
	7	STOR	-	-	1	1.05	1.17	-
	8	PVOL	-	0.87	1	1.15	1.3	-
Personal	9	ACAP	1.42	1.19	1	0.85	0.71	-
	10	PCAP	1.34	1.15	1	0.88	0.76	-
	11	PCON	1.29	1.12	1	0.9	0.81	-
	12	AEXP	1.22	1.1	1	0.88	0.81	-
	13	PEXP	1.19	1.09	1	0.91	0.85	-
	14	LEXP	1.2	1.09	1	0.91	0.84	-
Proyecto	15	TOOL	1.17	1.09	1	0.9	0.78	-
	16	SITE	1.2	1.09	1	0.93	0.86	0.8
	17	SCED	1.43	1.14	1	1	1	-

Utilizamos la fórmula del esfuerzo estimado:

$$PM_{estimado} = PM_{nominal} * \prod_{t=1}^{17} EM_j$$

$$PM_{estimado} = 18 * (1 * 1 * 0.73 * 1 * 1 * 1 * 1 * 1 * 0.87 * 0.85 * 0.76 * 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1 * 1)$$

$$PM_{estimado} = 7.3$$

El resultado indica que se necesita 12 personas trabajando por un mes para desarrollar el proyecto. Este valor sirve para hallar la estimación de duración del proyecto.

5.1.4 Estimación de duración del proyecto

$$Destimada = [C * PM_{estimado}] * [SCED\%/100]$$

Donde:

Destimada = Duración estimada del proyecto

SCED = Porcentaje de planificación de desarrollo del software con tiempos comprometidos.

D = Exponente de escalamiento ($D = 0.28 + 0.2 * (B - 0.91)$)

PMestimado = Esfuerzo estimado del proyecto

C = Coeficiente de planificación (C=3.67)

Mediante la fórmula primero calculamos D:

$$D = (0.28 + 0.2 * (0.94 - 0.91))$$

$$D = 0.286$$

Luego reemplazamos en la fórmula de Destimada:

$$Destimada = [3.67 * 7.3^{0.286}] * [100/100]$$

$$Destimada = 6.4 \text{ mes}$$

Considerando que el tiempo de desarrollo del proyecto debe ser más corto entonces el tiempo de desarrollo debe de reducirse.

El resultado nos indica que la duración del desarrollo es de aproximadamente 7 meses.

5.1.5 Costo de desarrollo

Por ultimo calculamos el costo de desarrollo del sistema, esta fórmula esta descrita por:

CostoDesarrollo

$$= \text{NúmeroProgramadores} * \text{Destimada} / \text{NumeroProgramadores} \\ * \text{SalarioProgramador}$$

Tomando en cuenta que el costo de contratar programadores es de 600 dolares que son entre 4200 bs por programador.

Entonces:

$$\text{CostoDesarrollo} = 2 * 4200 * 6$$

$$\text{CostoDesarrollo} = 50400 \text{ Bs}$$

Se tiene en consideración que el costo de desarrollo disminuye al hacer uso de librerías, framework y servicios ya disponibles.

5.1.6 Costo de implementación

El costo de implementación se apoya en los requerimientos ya existentes, solo se requiere la configuración del servidor y la adquisición de un hosting.

El proyecto usa el hosting gratuito de firebase denominado firebase hosting y el servidor para la base de llamado firestore que cuenta con un plan gratuito.

También se debe de agregar el costo de adquisición del dominio web que es de 90 bolivianos.

Por lo tanto:

$$\text{Costo de Implementación} = 90 \text{ Bs}$$

5.1.7 Costo de elaboración

Estos gastos están presentes en las etapas o fases que requiere la AUP, entonces se tiene:

Tabla 41: Costo de Elaboración

Detalle	Gasto en Bs
Papelería	80 Bs
Otros	161 Bs
Costo Total Final	724 Bs

5.1.8 Costo total de elaboración del proyecto

Para determinar el costo total del proyecto seguimos la siguiente tabla:

Tabla 42: Costo total del proyecto

Detalle	Gasto en Bs
----------------	--------------------

Costo de desarrollo	50400Bs
Costo de implementación	90 Bs
Costo de elaboración	724 Bs
Costo Total Final	51214 Bs

CAPITULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Una vez concluido con el desarrollo de la aplicación web para la Unidad Educativa “Calama”, aplicando las metodologías de desarrollo de software en base a los objetivos propuestos, se afirma que se han alcanzado los mismos.

Las conclusiones a las que se llega son:

- Se realizó el análisis de la información necesaria para el funcionamiento del sistema.
- Se estructuró y configuró una base de datos no relacional capaz soportar la cantidad de usuarios simultáneos.
- Se desarrolló un módulo de inscripción que optimiza en tiempo de creación de cuenta para los usuarios de la institución.
- Se desarrolló una interfaz web que organiza y sistematiza los contenidos y materiales curriculares y tecnológicos de la Unidad Educativa.
- Se desarrolló un nuevo proceso de generación de reportes basados en criterios de evaluación propios en educación.

6.2 RECOMENDACIONES

Las recomendaciones en cuanto a la aplicación web son las siguientes:

- Para futuras actualizaciones o modificaciones del sistema se recomienda utilizar el framework React en su versión 16.
- El servidor actual no cuenta con un servicio de almacenamiento de archivos propio.
- Actualmente la base de datos cuenta con un nivel de des normalización de un nivel y se recomienda realizar un segundo nivel de des normalización para mejorar el rendimiento de consultas a colecciones más recurrentes.

- Se recomienda que para futuras agregaciones de sistemas se utilice las transacciones de firebase para el correcto funcionamiento de operaciones de actualización y eliminación de documentos.
- En el caso de migración de la base de datos se sugiere considerar el soporte para más de 800 usuarios en el sistema y un máximo de 100 usuarios conectados simultáneamente.

BIBLIOGRAFIA

- Apaza, F. (2016). *Plataforma colaborativa de cursos masivos en línea*. (Tesis). Universidad Mayor de San Andrés. La Paz, Bolivia.
- Bermejo, M. (2015). *El Kanban*. Universitat Oberta de Catalunya. España.
- Booch, G., Rumbaugh, J. y Jacobson, I. (2005). *Unified Modeling User Guide, The, Second Edition*. EE. UU. Addison-Wesley Professional.
- Brechner, E. (2015). *Agile Project Management with Kanban*. EE. UU. Microsoft Press.
- Cabrera, A., Figeroa, R. y Solís, C. (2008). *Metodologías tradicionales vs Metodologías Ágiles*. Universidad Técnica Particular de Loja. Escuela de Ciencias en Computación.
- Dávila, M. (2013). *Análisis, diseño e implementación de un sistema de aula virtual para captación de personal en la empresa UnderMedia S. A.* (Tesis). Escuela Politécnica del Ejercito. Sangolqui, Ecuador.
- Facebook Inc (2020). Empezando con React. [es.reactjs.org](https://es.reactjs.org/docs/getting-started.html). Obtenido de <https://es.reactjs.org/docs/getting-started.html>.
- Firebase (2020). Cloud Firestore. Firebase. Obtenido de <https://firebase.google.com/docs/firestore?hl=es-419>.
- Foreman, S. (2017). *The LMS Guidebook: Learning Management Systems Demystified*. EE. UU. Association for Talent Development.
- Freeman, E. y Roboson, E. (2020). *Head First Design Patterns*. EE. UU. O'Reilly Media Inc.
- Ingeno, J. (2018). *Software architect's Handbook*. EE. UU. Packt Publishing.
- Maurer, R. (2012). *The Spirit of Kaizen: Creating Lasting Excellence One Small Step at a Time*. EE. UU. McGraw-Hill.

- Medinilla, A. Giné, A. y Gómez, E. (Sin año). *Manifiesto por el desarrollo Ágil de Software*. Recuperado de <http://agilmanifiesto.org.iso/es/principles.html>.
- Ministerio de Educación Culturas y Deportes. (2020). Reglamento Específico de Complementariedad de las Modalidades de Atención Presencial, a Distancia, Virtual y Semipresencial del Subsistema de Educación Regular. Minedu.gob.bo. Bolivia. <https://www.minedu.gob.bo/files/documentos-normativos/reglamentos/2020/edu-regular.pdf>
- Ministerio de Educación Culturas y Deportes. (2020). *Reglamento de Evaluación del Desarrollo Curricular Documento Preliminar*. Minedu.gob.bo. Bolivia. https://drive.google.com/file/d/1DDmnaFhJQ8DWQ2YnergEuBZ1DdjNOpkG/view?fbclid=IwAR2BTV-NQ8qYkp_rwFxFyA0ohHIKrmzCHUzOB60oNwGuOdKyfyeFLNB0ozY
- Moodle Pty Ltd (2020). *Development Moodle*. Obtenido de <https://docs.moodle.org/dev>.
- Mouse, J. (2011). MVC: Modelo, Vista y Controlador. Obtenido de <http://www.jcmouse.net/proyectos/mvc-modelo-vista-y-controlador-en-php>
- Mozilla (2020). *JavaScript*. Developer.mozilla.org. Obtenido de <https://developer.mozilla.org/es/docs/Web/JavaScript>.
- Palacios, J. (2016). *Kanban (III): Reuniones y roles en Kanban*. Obtenido de <https://jeronimopalacios.com/kanban>.
- Powel, B. (2014). *Real-Time UML Workshop for Embedded Systems, 2nd Edition*. EE. UU. Newnes.
- Pressman, R. (2010). *Ingeniería del software un Enfoque Práctico*, Séptima Edición. Mexico. McGraw-Hill Interamericana Editores, S. A. de C. V.
- Raffino, M. (2020). *Técnicas de Investigación*. Concepto.de. Obtenido de <https://concepto.de/tecnicas-de-investigacion>.
- Renteria, M. (2015). *Implementación de una plataforma virtual como estrategia metodológica que permita mejorar el rendimiento académico en el área de*

matemáticas. (Maestría). Universidad Pontificia Bolivariana. Medellín, Colombia.

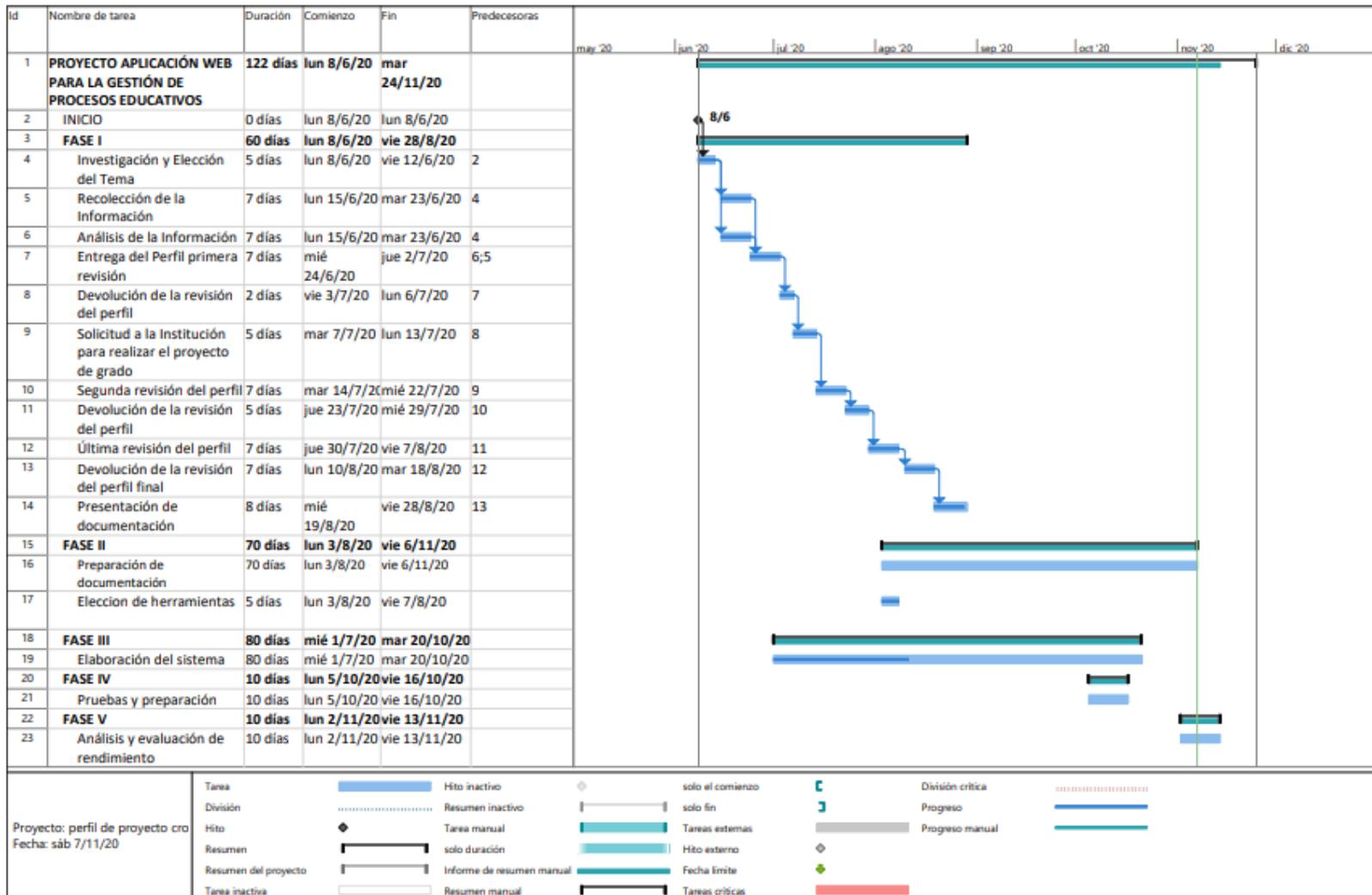
Sánchez, J. (2015). *Pruebas de Software. Fundamentos y Técnicas*. Obtenido de http://oa.upm.es/40012/1/PFC_JOSE_MANUEL_SANCHEZ_PENO_3.pdf

Sommerville, I. (2005). *Ingeniería del Software*. Séptima Edición. España. Pearson Education Limited.

Stellman, A. y Greene, J. (2004). *Learning Agile*. EE. UU. O'reilly Media Inc.

ANEXOS

ANEXO A: CRONOGRAMA



ANEXO B: REGLAS DE SEGURIDAD DE LA BASE DE DATOS

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /cursos/{idCurso}/inscripciones/{idEstudiante} {
      allow write: if request.auth != null &&
        (get(/databases/{database}/documents/users/{request.auth.uid}).data.rol ==
'estudiante');
      allow read: if true;
    }
    match /cursos/{icCurso} {
      allow write: if request.auth != null &&
        (request.auth.uid == resource.data.idDocente ||
        get(/databases/{database}/documents/users/{request.auth.uid}).data.rol ==
'docente');
      allow read: if true;
    }
    match /users/{idUser} {
      allow write: if request.auth != null;
      allow read: if true;
    }
    match /horarios/{idHorario} {
      allow write: if request.auth != null;
      allow read: if true;
    }
    match /comunicados/{idComunicado} {
      allow write: if request.auth != null;
      allow read: if true;
    }
    match /trimestres/{idTrimestre} {
      allow write: if request.auth != null;
```

```
    allow read: if true;
  }
  match /contenidos/{idContenido} {
    allow write: if request.auth != null;
    allow read: if true;
  }
  match /preguntas/{idPregunta} {
    allow write: if request.auth != null;
    allow read: if true;
  }
  match /preguntasCursos/{idCurso} {
    allow write: if request.auth != null;
    allow read: if true;
  }
  match /respuestas/{idRespuesta} {
    allow write: if request.auth != null;
    allow read: if true;
  }
  match /respuestasCursos/{idCurso} {
    allow write: if request.auth != null;
    allow read: if true;
  }
  match /aportes/{idAporte} {
    allow write: if request.auth != null;
    allow read: if true;
  }
  match /aportesCursos/{idCurso} {
    allow write: if request.auth != null;
    allow read: if true;
  }
  match /notas/{idNota} {
```

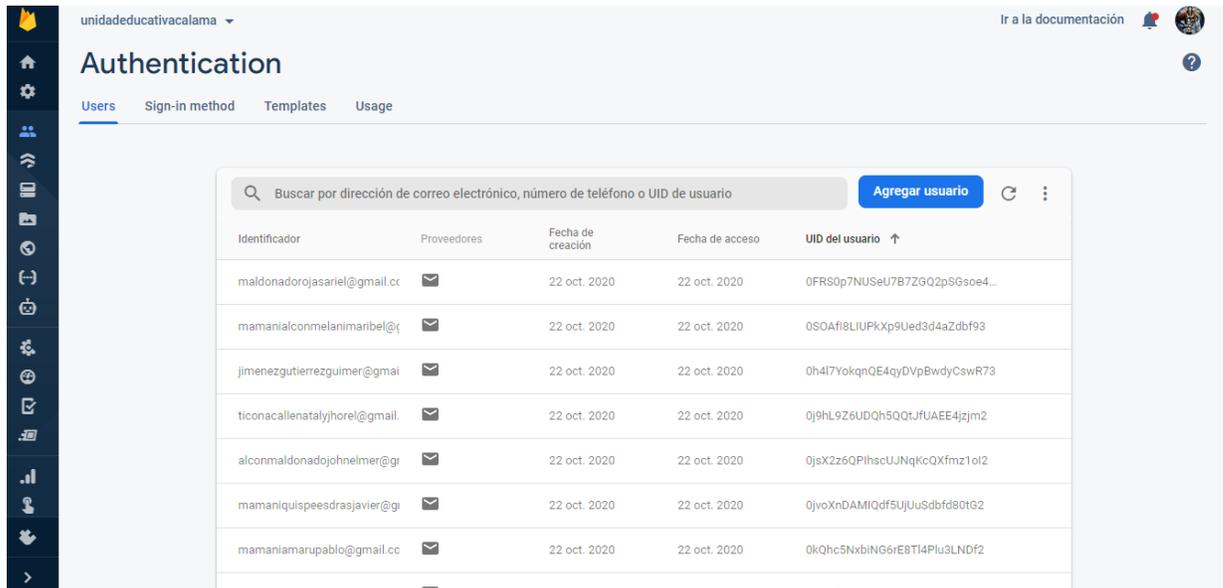
```

    allow write: if request.auth != null;
    allow read: if true;
}
match /notasCursos/{idCurso} {
    allow write: if request.auth != null;
    allow read: if true;
}
match /tareas/{idTarea} {
    allow write: if request.auth != null;
    allow read: if true;
}
match /tareasCursos/{idCurso} {
    allow write: if request.auth != null;
    allow read: if true;
}
match /claves/{codigo} {
    allow write: if request.auth != null;
    allow read: if true;
}
match /{path=**}/preguntas/{idPregunta}{
    allow read, write;
}
match /{path=**}/aportes/{idAporte}{
    allow read, write;
}
match /{path=**}/notas/{idNota}{
    allow read, write;
}
match /{path=**}/tareas/{idTarea}{
    allow read, write;
}

```

```
match /{path=**}/respuestas/{idRespuesta}{
  allow read, write;
}
match /{path=**}/participaciones/{idEstudiante}{
  allow read, write;
}
}
}
```

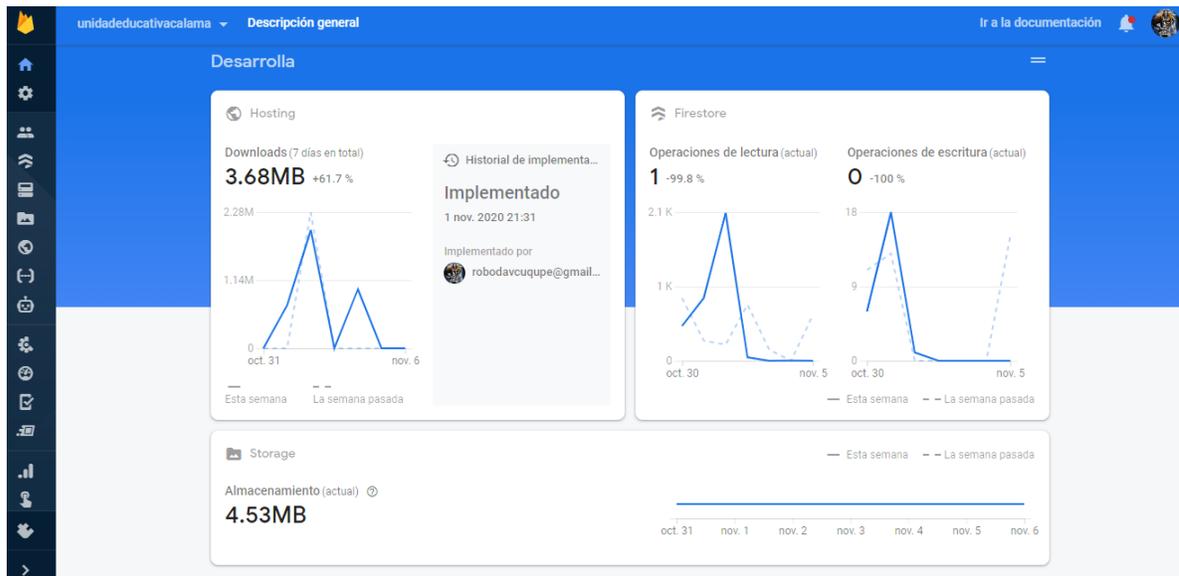
ANEXO C: PANEL DE CONTROL DE ACCESOS DE USUARIOS



The screenshot displays the 'Authentication' control panel for 'unidadeducativacalama'. The main heading is 'Authentication', with sub-tabs for 'Users', 'Sign-in method', 'Templates', and 'Usage'. The 'Users' tab is active, showing a search bar and a table of users. The search bar contains the text 'Buscar por dirección de correo electrónico, número de teléfono o UID de usuario'. A blue button labeled 'Agregar usuario' is located to the right of the search bar. The table lists seven users, all created on 22 oct. 2020 and accessed on the same date. Each user entry includes an email address, a provider icon, and a unique User ID.

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID del usuario ↑
maldonadorojasariel@gmail.cc	✉	22 oct. 2020	22 oct. 2020	0FRS0p7NUSseU7B7ZGQ2pSGsoe4...
mamanielconmelanimaribel@t	✉	22 oct. 2020	22 oct. 2020	0SOAf18LIUPkXp9Ued3d4aZdbf93
jimenezgutierrezguimer@gmail	✉	22 oct. 2020	22 oct. 2020	0h4I7YokqnQE4qyDVpBwdyCswR73
ticonacallenatalyjhorel@gmail.	✉	22 oct. 2020	22 oct. 2020	0j9hL9Z6UDQh5QQtJfUAEE4jzm2
alconmaldonadojohnelmer@gr	✉	22 oct. 2020	22 oct. 2020	0jsX2z6QPihscUJNqKcQXfmz1ol2
mamaniquispeedrasjavier@gr	✉	22 oct. 2020	22 oct. 2020	0jvoXnDAMIQdf5UJuuSdbfd80tG2
mamaniamarupablo@gmail.cc	✉	22 oct. 2020	22 oct. 2020	0kQhc5NxbiNG6rE8T14Plu3LNDf2

ANEXO D: PANEL DE CONTROL HOSTING Y BASE DE DATOS



ANEXO E: PANEL DE CONTROL DATOS DE LA BASE DATOS

The screenshot displays the Google Cloud Firestore console interface. At the top, the breadcrumb navigation shows 'unidadeducativacalama > users > 0FRS0p7NUseU...'. The main area is divided into three columns:

- Left Column:** A collection list for 'unidadeducativacalama' with a '+ Iniciar colección' button. Collections include 'aportesCursos', 'claves', 'comunicados', 'contenidos', 'cursos', 'horarios', 'notasCursos', 'preguntasCursos', 'respuestasCursos', 'tareasCursos', 'trimestres', and 'users' (highlighted).
- Middle Column:** A document list for the 'users' collection with a '+ Agregar documento' button. The selected document ID is '0FRS0p7NUseU7B7ZGQ2pSGsoe4p2'. Below the ID, a list of document IDs is visible, including '0S0AfI0LIUPkXp9Ued3d4aZdbf93', '0h417YokqnQE4qyDVpBwdyCswR73', '0j9hL9Z6UDQh5QQtJfUAEE4jzm2', '0jsX2z6QPIhscUJNqKcQXfmz1oI2', '0jvoXnDAMIQdf5UjUuSdbfd80tG2', '0kQhc5NxbiNG6rE8T14P1u3LNDf2', '0kUSifQ8pZStZW181B8U0bnsEDH2', '0kv1CkIEDubpvhjMpnLUK1P3Zmj1', '0l30MyAuU9W1QvYChrKsIsAPjRV2', '0r4puRuQAEed2LyrEj2nTRQ7yPOT2', '1500IkBtPEQ0p4QVehGmMHKeBhJ3', '15Zi5z92NnXjvFWEYtAnFSnfeQx2', and '19vA3hMsGnTTCR3STHwWckv8AYE3'.
- Right Column:** A document view for the selected document with a '+ Iniciar colección' button and a '+ Agregar campo' button. The document fields are: 'apellidoMaterno: "ROJAS"', 'apellidoPaterno: "MALDONADO"', 'codigo: "6B"', 'curso: "6 Sexto B"', 'email: "maldonadorojasariel@gmail.com"', 'fecha: 22 de octubre de 2020 a las 18:32:35 UTC-4', 'nombre: "ARIEL"', 'rol: "estudiante"', and 'uid: "0FRS0p7NUseU7B7ZGQ2pSGsoe4p2"'. A vertical scrollbar is visible on the right side of this column.

ANEXO F: MANUAL DE USUARIO

Manual de Usuario Docente

Proyecto: Aplicación Web para la Gestión de Procesos Educativos

Revisión 1

Autor: David Gonzalo Cusi Quispe



Aplicación Web Calama

Tabla de Contenido

- 1 Descripción
- 2 Instalación
- 3 Ingreso a la aplicación web
- 4 Iniciar Sesión
- 5 Crear Curso
- 6 Ver Cursos Creados
- 7 Eliminar un Curso
- 8 Agregar Trimestres a Cursos
- 9 Agregar un video de Youtube
- 10 Editar un contenido
- 11 Compartir un documento de Google Drive
- 12 Opciones de contenidos
- 13 Opción Ver Contenido archivo
- 14 Generar lista de calificaciones

El Alto, noviembre del 2020

1 Descripción

El sistema se encarga de gestionar la información referente a los contenidos curriculares y herramientas utilizadas por los docentes y estudiantes de la Unidad Educativa Calama. La información que gestiona el sistema va desde documentos digitales almacenados en google drive, enlaces a grupos de whatsapp, videos compartidos en youtube y enlaces a reuniones de zoom.

Toda la información cargada en el sistema estará organizada en la aplicación web con la siguiente dirección url:

<https://unidadeducativacalama.web.app/>

El acceso al sistema se podrá realizar mediante un correo electrónico registrado en el sistema.

1.1 Requerimientos

Los requisitos para acceder a la aplicación web son:

- Tener un dispositivo electrónico con conexión a internet
- Un Navegador web Chrome, Firefox, Edge u otro navegador actualizado.

2 Instalación

El proyecto utiliza principalmente como entorno de desarrollo el framework React y la base de datos Firestore de firebase.

Para realizar la instalación del sistema en otro servidor se tiene que clonar el proyecto del siguiente repositorio:

<https://github.com/davigox/unidadeducativacalama>

Y ejecutar los siguientes en la terminal de comandos:

Para clonar el proyecto

- `git clone https://github.com/davigox/unidadeducativacalama.git`
- `cd unidadeducativacalama`

Por ultimo instalamos todas las librerías del proyecto

- `npm install`

3 Ingreso a la aplicación web

Paso 1: Ingresar en el navegador la siguiente dirección web:

<https://unidadeducativacalama.web.app/>

Al ingresar la dirección nos aparecerá la siguiente ventana principal



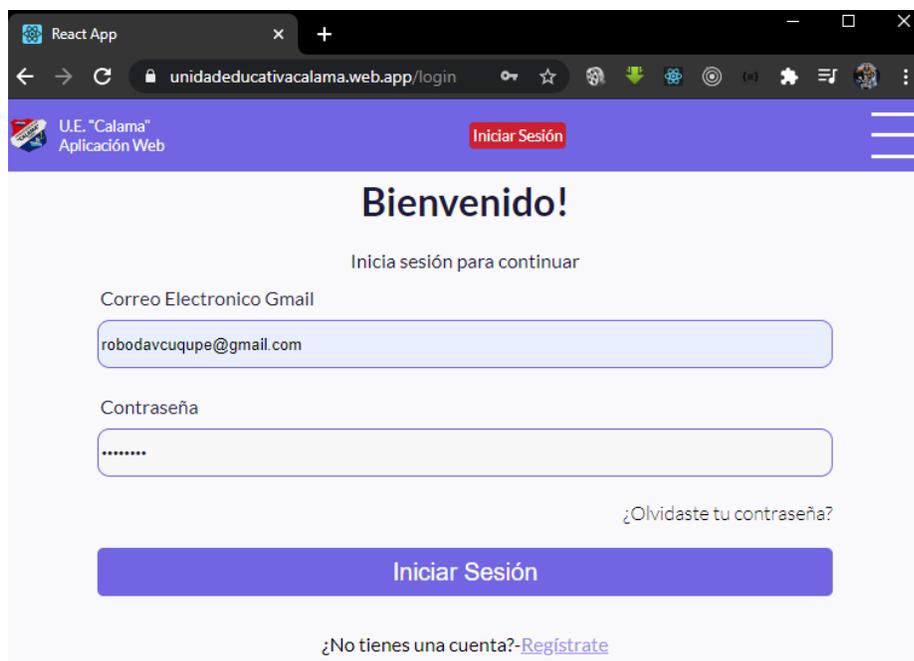
4 Iniciar Sesión

Para iniciar sesión en el sistema el usuario previamente debe tener una cuenta creada en el sistema.

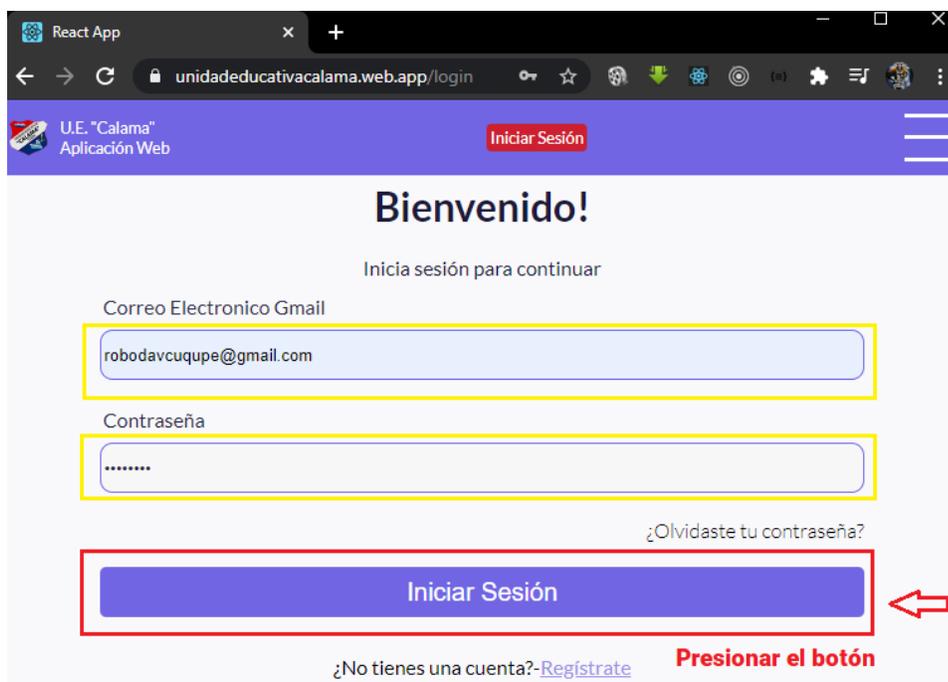
Paso 1: Presionar el botón de Iniciar Sesión.



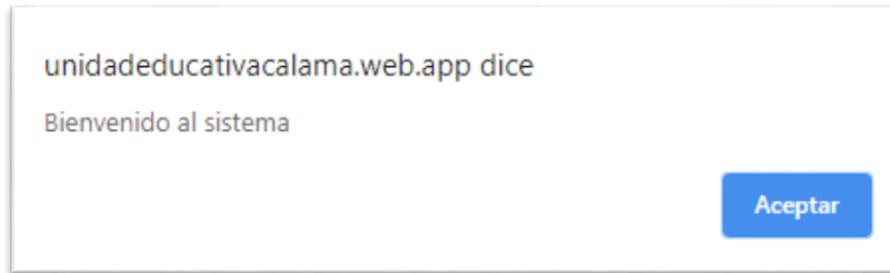
Paso 2: Una vez presionado el botón de Iniciar Sesión se mostrará un formulario de inicio de sesión en donde se debe ingresar su correo electrónico y la contraseña correspondiente.



Paso 3: Por ultimo debe presionar el botón de Iniciar Sesión.



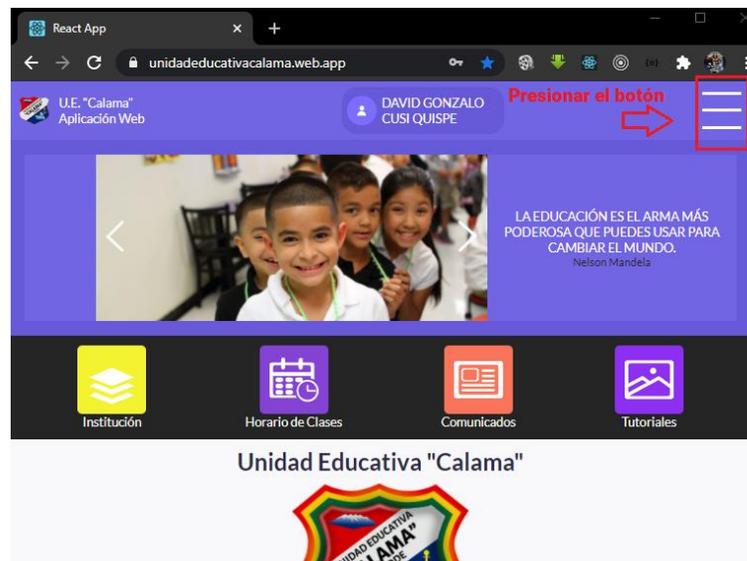
Una vez presionado el botón el sistema le mostrará el siguiente mensaje:



5 Crear Curso

Los pasos que se debe seguir para crear un curso son los siguientes.

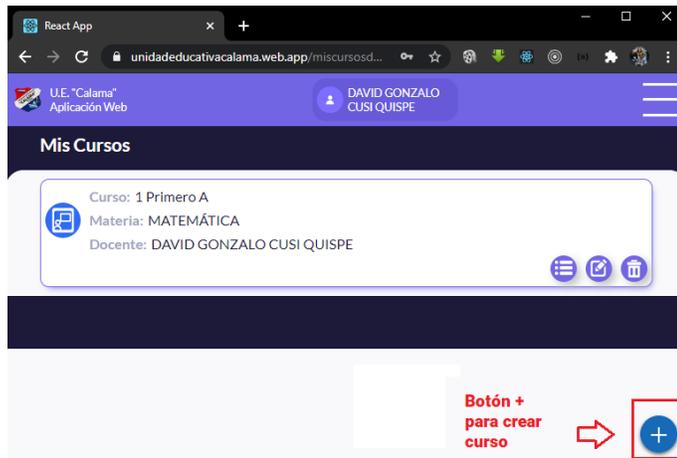
Paso 1: Presionar en el botón de Menú.



Paso 2: Seleccionar la opción de Mis Cursos.

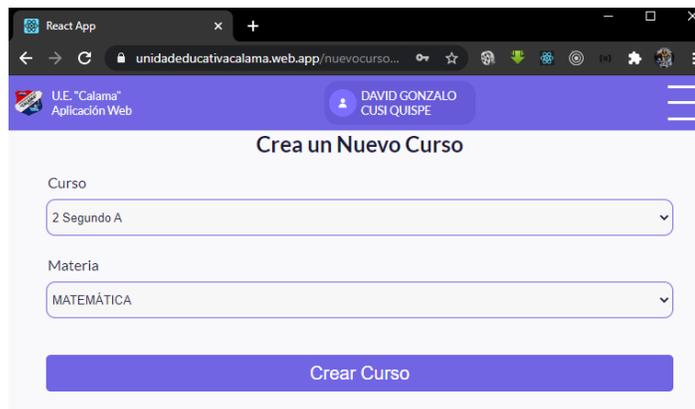


Paso 3: Presionar el botón + para crear el curso.

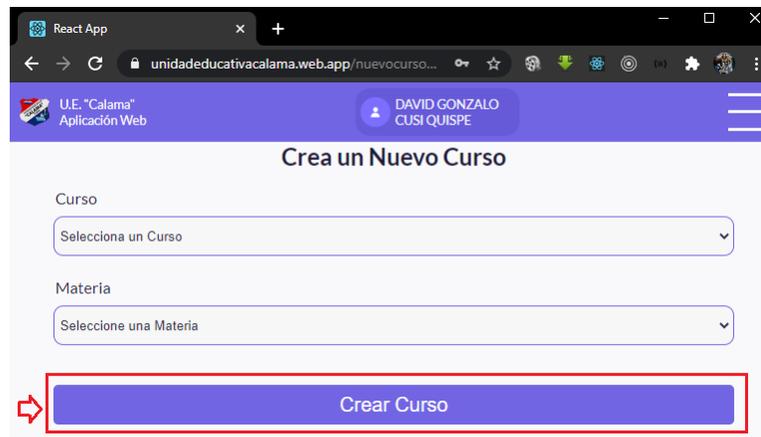


Paso 4: El sistema mostrará un formulario donde se debe escoger el curso y la materia que se creará.

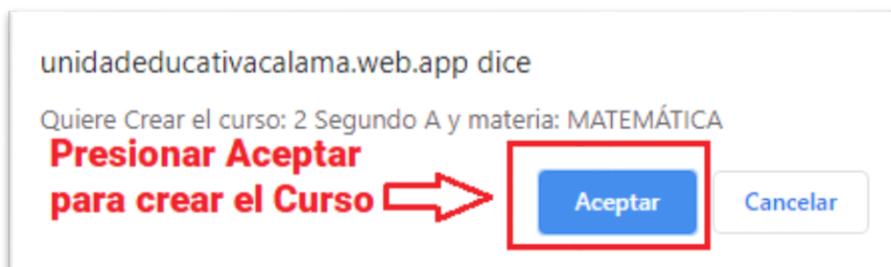
Para el ejemplo de este manual se creará el curso de 2do A de Secundaria con la materia de Matemática.



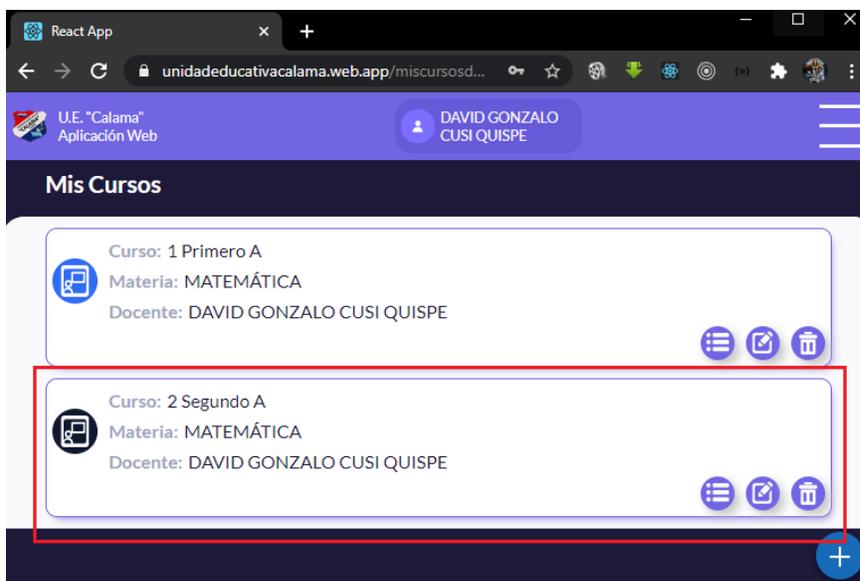
Paso 5: Presionar el botón de Crear Curso.



Paso 6: Como último paso debemos esperar a que el sistema no muestre el mensaje de confirmación.

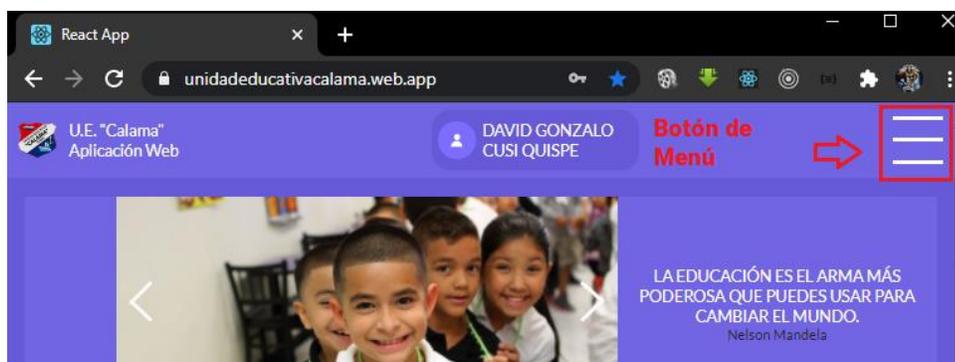


Una vez hecho el paso 6 el sistema nos mostrará una lista con los curso que tenemos.



6 Ver Cursos Creados

Paso 1: Seleccionar el botón de Menú.



Paso 2: Seleccionar la opción de Mis Cursos.



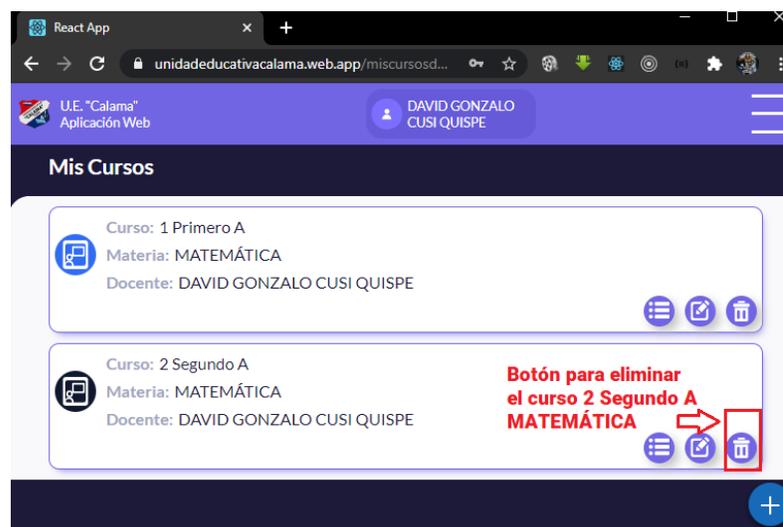
Luego de hacer el paso 2 se mostrarán todos los cursos que tiene creados.

7 Eliminar un Curso

En la lista de Cursos cada curso tiene un botón de eliminación como se puede observar en la imagen.

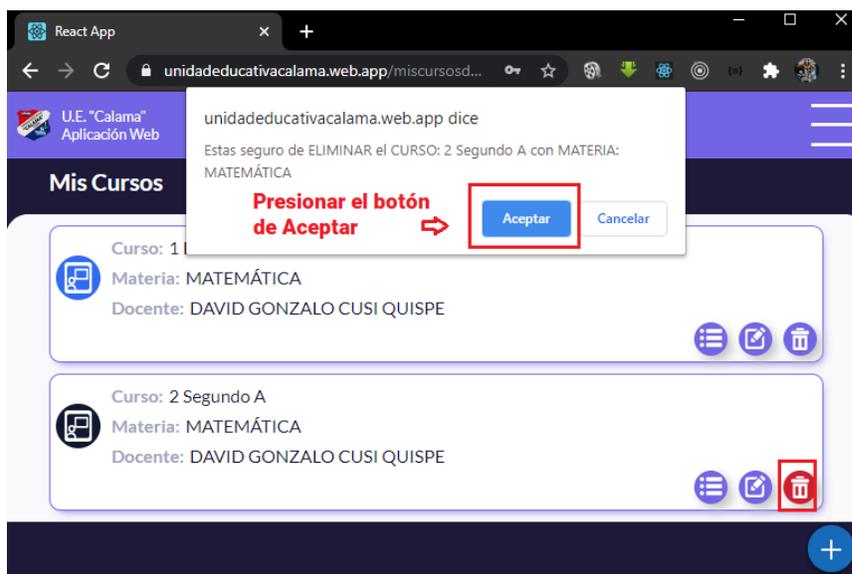
Paso 1: Presionar el botón de eliminación de un curso.

Para el ejemplo de este manual eliminaremos el curso de 2 Segundo A MATEMÁTICA.

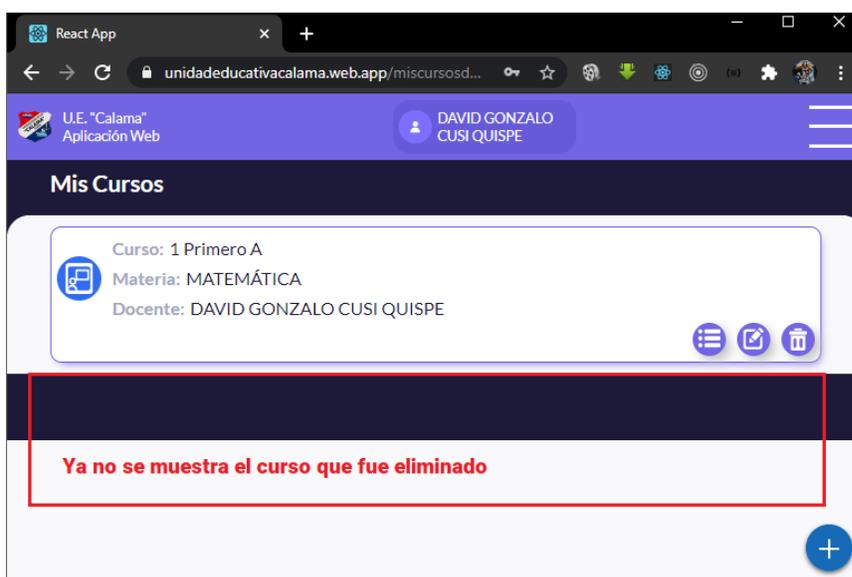


Al presionar el botón no aparecerá un mensaje de confirmación indicando el curso que se eliminará.

Paso 2: En el mensaje de confirmación indicando el curso que se eliminará. Presionamos el botón de **Aceptar**.

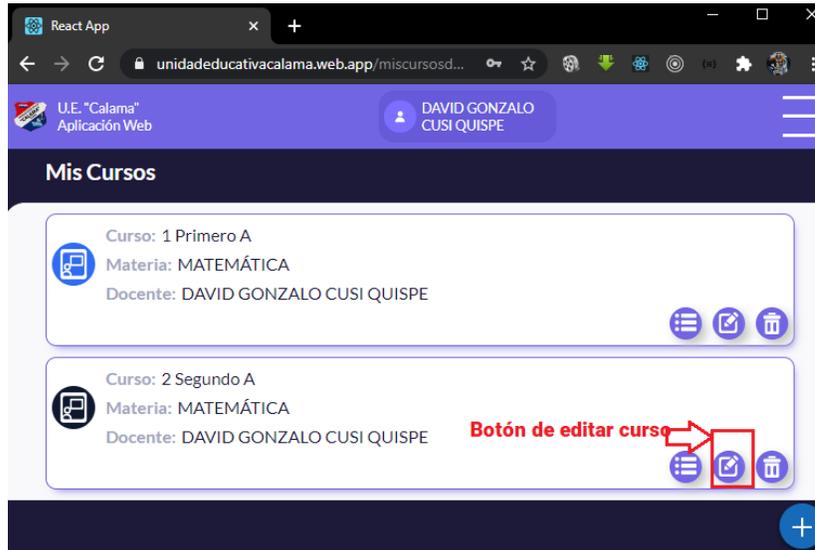


Por último, el sistema eliminará el curso y lo quitará de la lista de cursos.

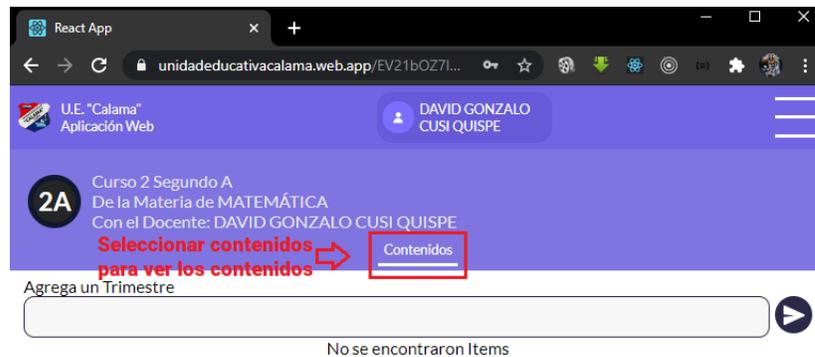


8 Agregar Trimestres a Cursos

Paso 1: Elegir la opción de editar Curso como se muestra en la siguiente imagen.

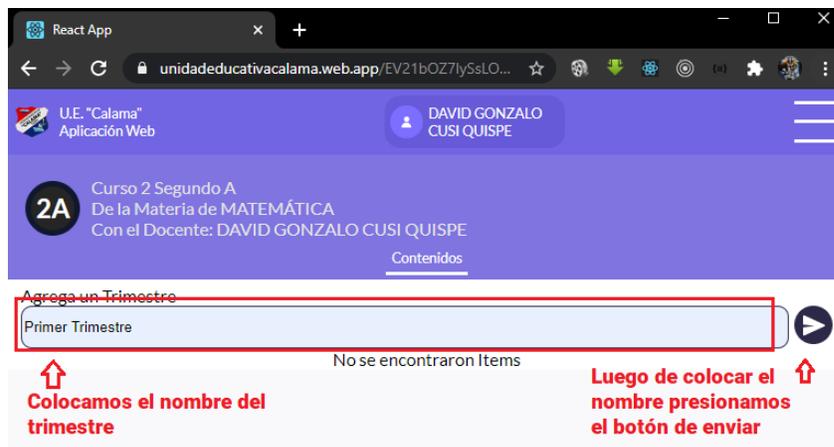


Paso 2: Seleccionar la opción de **contenidos**.

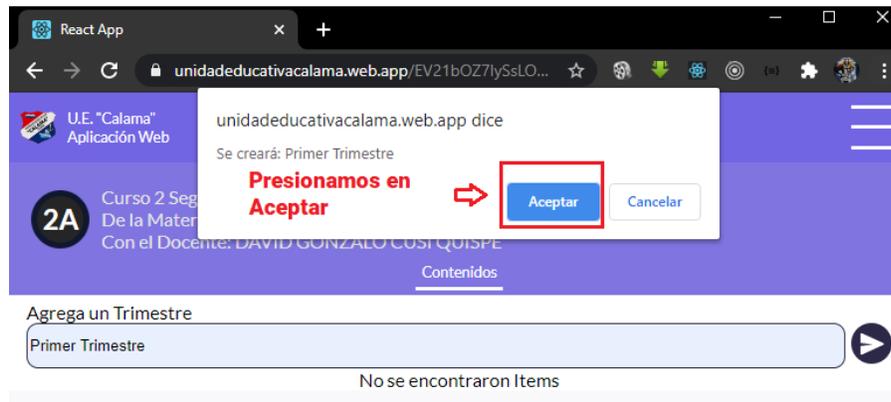


Paso 3: Llenar el formulario para crear un trimestre.

Se recomienda llenar el formulario con el nombre de Primer Trimestre.

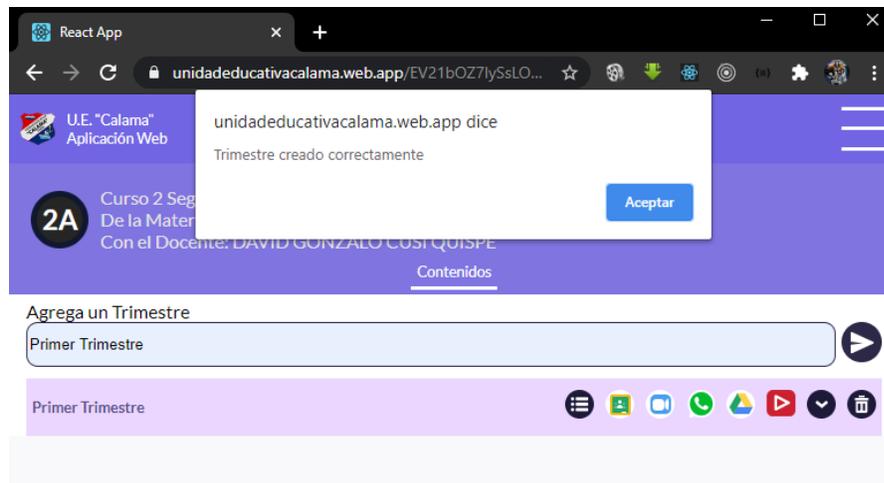


Paso 4: Una vez presionado el botón se nos mostrará un mensaje de confirmación que debemos de **aceptar**.



Una vez hecho el paso 4 el sistema nos mostrará el trimestre creado con un mensaje de confirmación.

Para agregar el Segundo Trimestre en el curso se repiten los pasos 3 y 4.

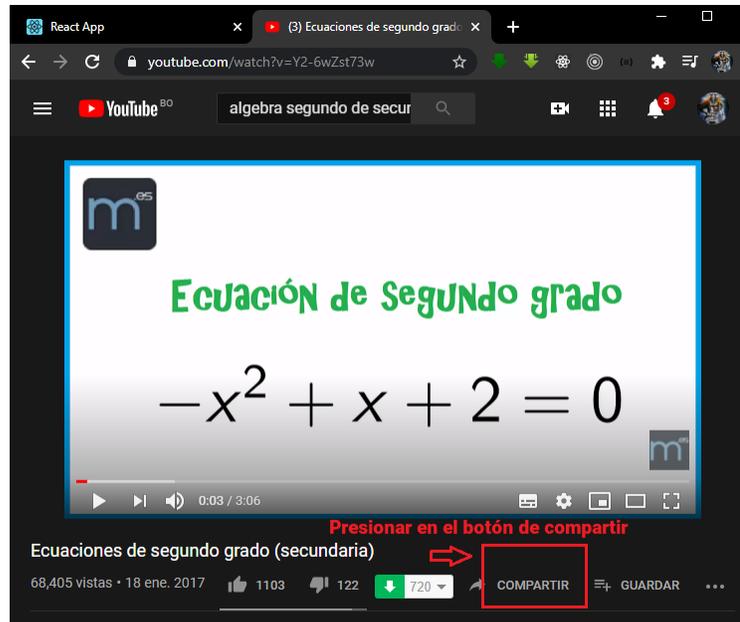


9 Agregar un video de Youtube

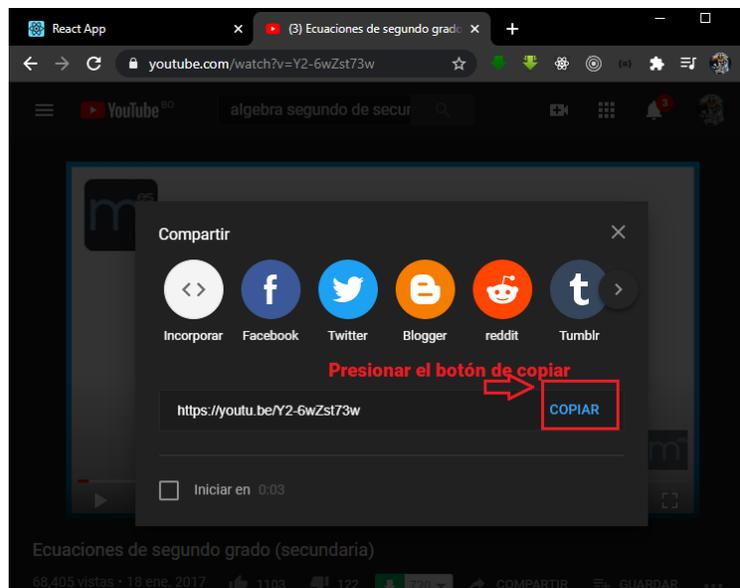
Una vez creado el trimestre podemos agregar o compartir un video de youtube referido al curso y la materia.

Paso 1: Presionar el botón de compartir del video de Youtube previamente.

Para el ejemplo del manual compartiremos un video de Algebra de Youtube de la siguiente manera:

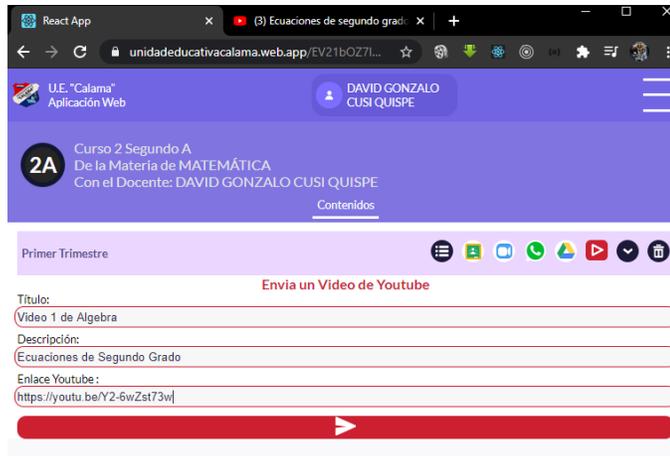


Paso 2: Presionar el botón de copiar.

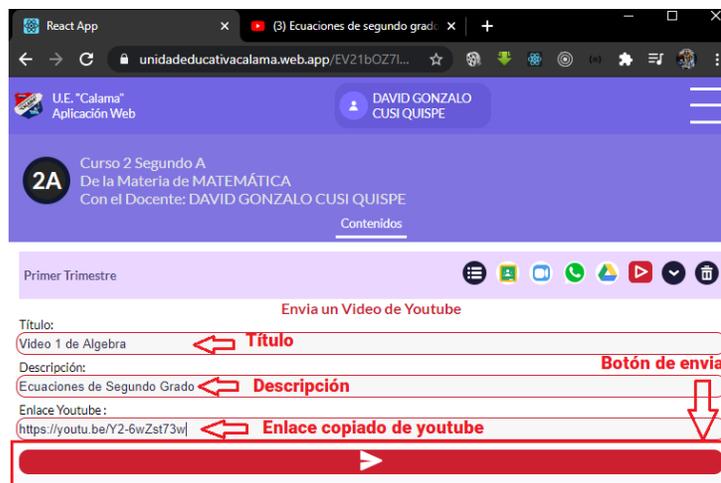


Paso 3: Seleccionar en el sistema el icono de youtube y pegamos el enlace copiado previamente.

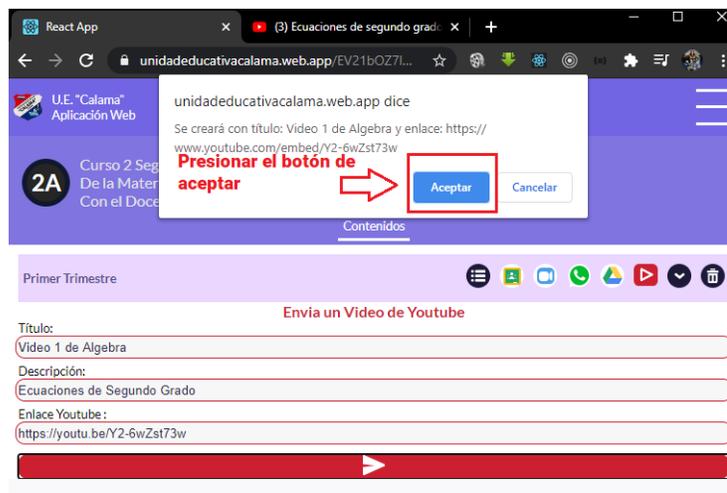
Adicionalmente también llenamos los campos de Título y descripción del video.



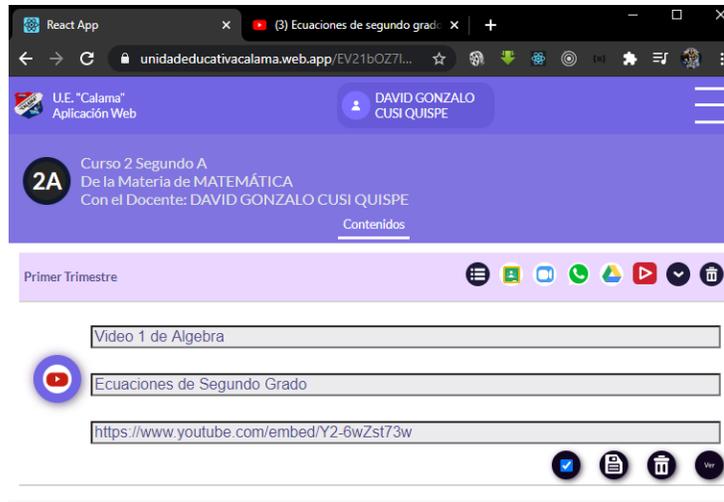
Paso 4: Presionamos el botón de **Enviar**.



Paso 5: Como último paso aceptamos el mensaje de confirmación del sistema.



Una vez hecho el paso 5 los estudiantes de Segundo de Secundaria podrán encontrar el video compartido de youtube en la aplicación de la Enidad Educativa



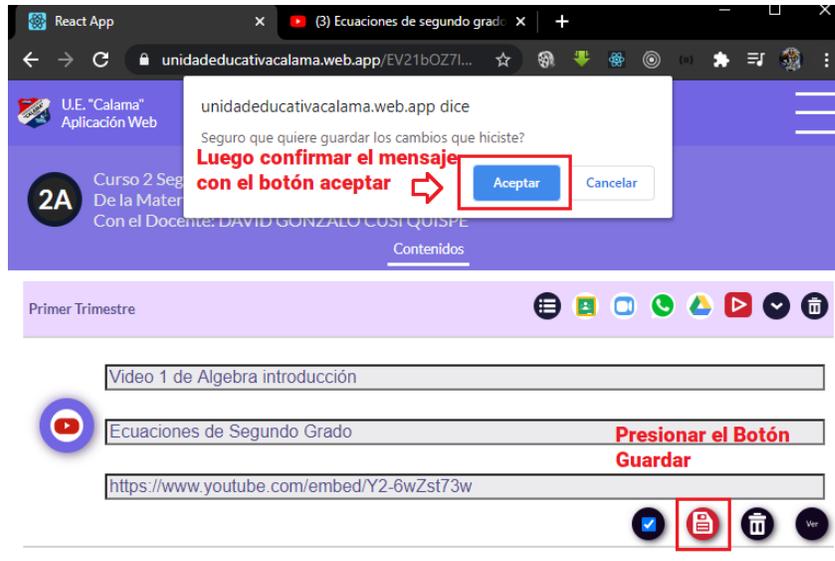
10 Editar un contenido

Si se hubiera cometido un error con el título descripción del video podemos cambiarlo y luego presionando en el botón de guardar.

Paso 1: Editar el título o descripción.



Paso 2: Presionar el botón de guardar y confirmar el mensaje.



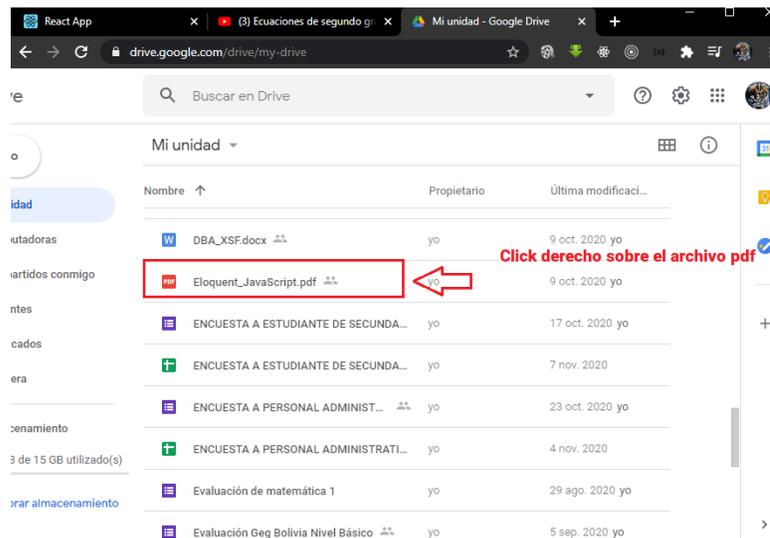
11 Compartir un documento de Google Drive

Para compartir documentos de Google Drive necesitamos cambiar antes los permisos de visualización para que los estudiantes no tengan problemas al acceder al documento.

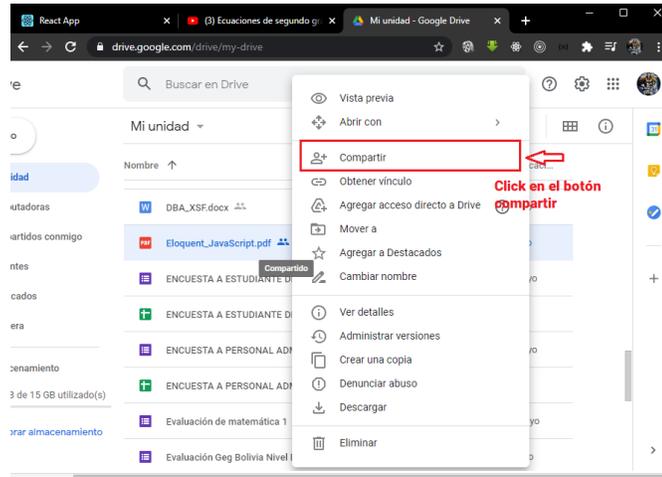
Los documentos que se pueden compartir son: Archivos Pdf, Documentos Word, Excel, Presentaciones, audios y formularios.

Paso 1: Preparar el documento de Google Drive. Hacemos Click derecho sobre el documento que queremos compartir.

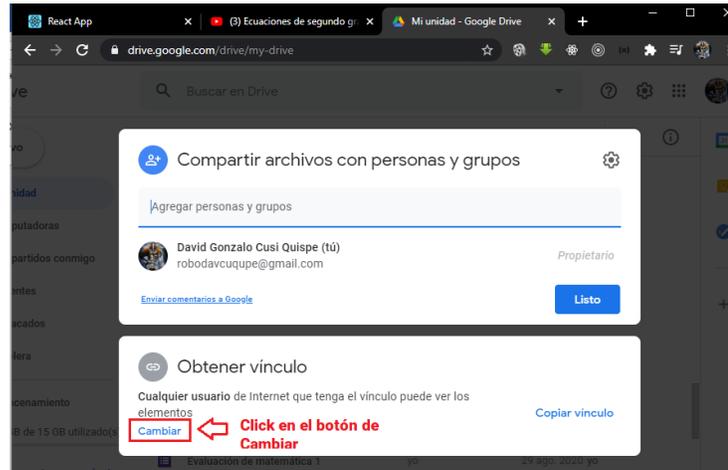
Para este ejemplo usaremos un texto pdf de Drive.



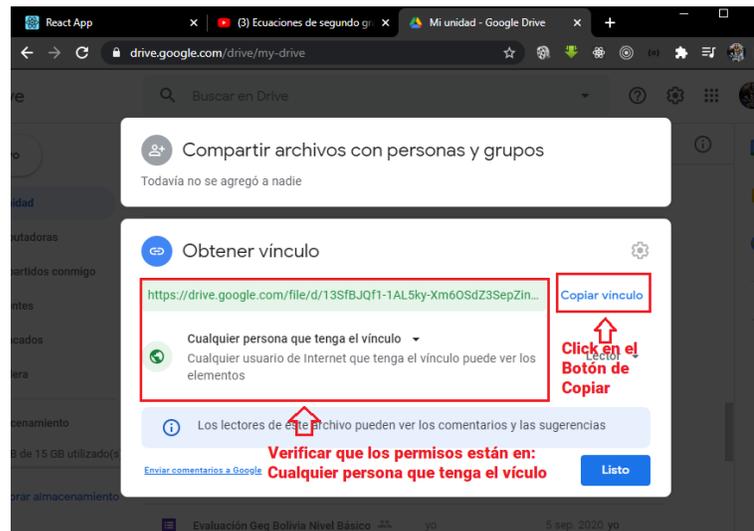
Paso 2: Seleccionar la opción de compartir.



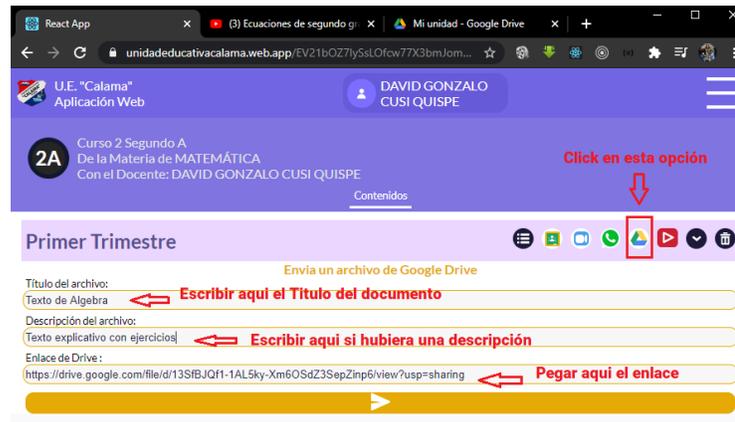
Paso 3: Click en el botón de Cambiar.



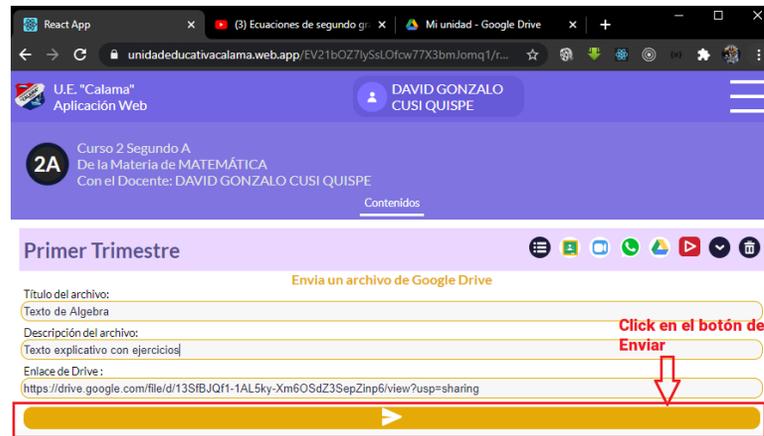
Paso 4: Click en el botón de Copiar vínculo.



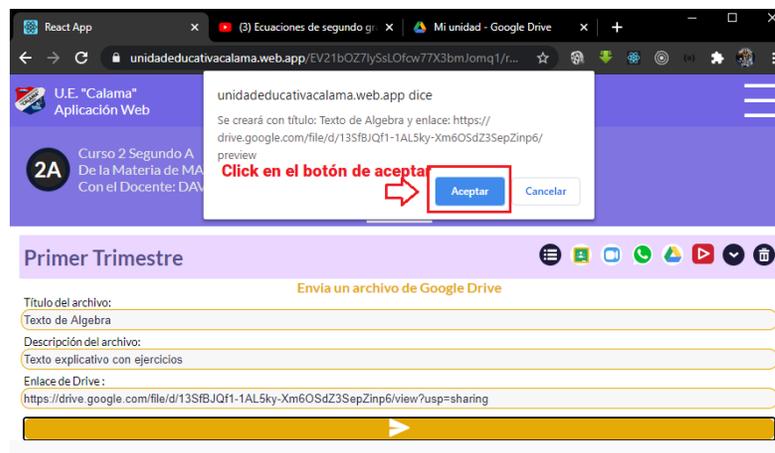
Paso 5: Pegar el enlace en el formulario de Compartir un archivo de Drive.



Paso 6: Enviar el documento con el botón de enviar.



Paso 7: Aceptar los mensaje de confirmación.



Los procedimientos para compartir una reunión Zoom o una Clase Classroom son los mismos teniendo como **paso importante el copiar correctamente los enlaces requeridos.**

12 Opciones de contenidos

Una vez tengamos contenidos cargados en el sistema se habilita las opciones de:

- Ver el contenido y su correcto funcionamiento
- Eliminar el contenido en caso de ser requerido.
- Bloquear el contenido para que los estudiantes no puedan entregar trabajos acerca del contenido

Al seleccionar un contenido podemos ver:

- Preguntas que hicieron los estudiantes.
- Las respuestas que hicieron los estudiantes.
- Los aportes como pequeños comentarios que de los estudiantes
- Las tareas entregadas por los estudiantes si es que hubieran.

Como docente uno puede:

- Plantear preguntas acerca del contenido.
- Si se considera que las preguntas, respuestas, aportes y tareas entregadas por los estudiantes son meritorios a una calificación se puede valorar con una calificación desde 1 a 100 puntos.

13 Opción Ver Contenido archivo

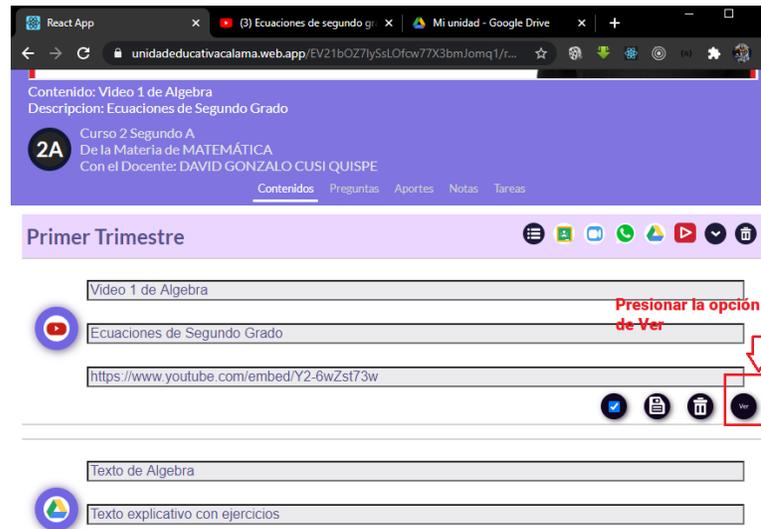
Si se tiene un documento o video compartido para los estudiantes se les habilita las opciones de:

- Ver y hacer preguntas (preguntas formuladas para todo el curso acerca del contenido).
- Ver y hacer respuestas (respuestas a preguntas hechas por otros estudiantes o el docente).
- Ver y hacer aportes (conocimientos cortos que el estudiante investigó).
- Ver y hacer Notas (anotaciones que el estudiante considere que son importantes relacionado con el contenido).
- Ver y entregar trabajos en documentos de Google Drive.

Como docente se tiene la posibilidad de ver y calificar cada pregunta, respuesta, aporte, nota, o tarea entregada por los estudiantes.

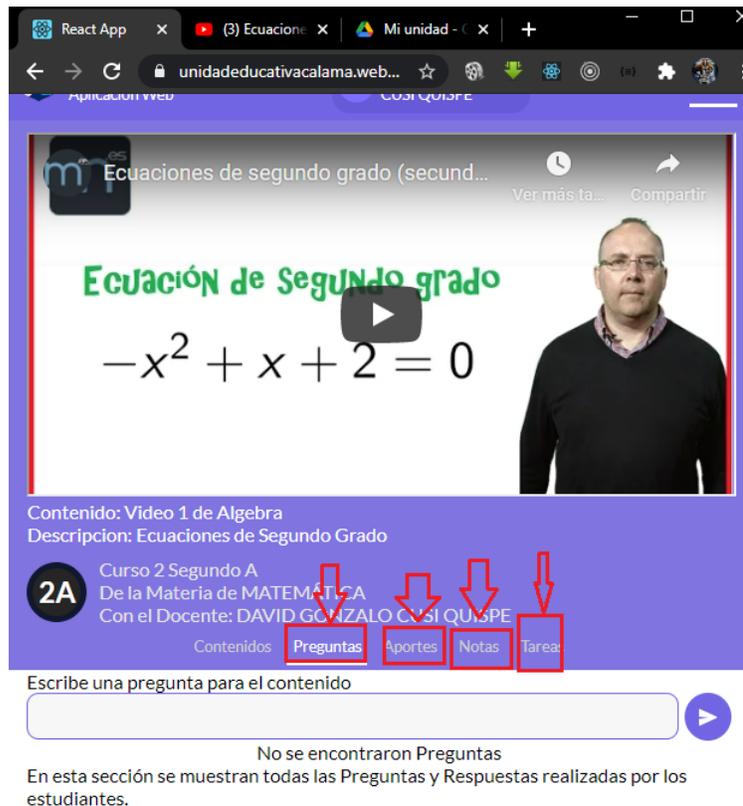
Los pasos para realizar esto son dos:

Paso 1: Presionar la opción de **ver** en el contenido.



Una vez hecho el paso 1 el sistema cargara en la parte superior el contenido con todas sus opciones.

Paso 2: Presionar la una de las opciones Preguntas, Aportes, Notas o Tareas.



Uno puede escoger cualquiera de las opciones para revisar los contenidos generados por los estudiantes.

14 Generar lista de calificaciones

Si se puso calificaciones a los estudiantes que crearon contenidos el sistema es capaz de generar un cuadro de evaluación con todos los estudiantes que fueron calificados por el docente.

Estas calificaciones son calculadas por trimestres y tomando en cuenta las las dimensiones de Ser/decidir, Saber y Hacer:

- Las preguntas realizadas por los estudiantes son consideradas en la dimensión del Hacer por 45 puntos.
- Las respuestas realizadas por los estudiantes son consideradas en la dimensión del Saber por 45 puntos.
- Los aportes realizados por los estudiantes son considerados en la dimensión del Saber por 45 puntos.
- Las tareas entregadas por los estudiantes son consideradas en la dimensión del Hacer por 45 puntos
- Las notas realizadas por los estudiantes son consideradas en la dimensión del Ser/Decidir por 10 puntos.

Nota. - Para que el sistema realice el cálculo de notas el docente debe previamente dar las calificaciones a los estudiantes.

Los pasos para generar las calificaciones son los siguientes:

Paso 1: Presionar la opción de lista del trimestre.



Paso 2: Una vez visualizando la lista podemos generar un archivo en formato pdf.

React App x (5) Ecuacione... x Mi unidad - G... x ListaCalificac... x +

unidadeducativacalama.web.app/estudianteslist/JrGoz... ☆

U.E. "Calama" Aplicación Web DAVID GONZALO CUSI QUISPE

Registro de participaciones

Curso: 1 Primero A
Materia: MATEMÁTICA
Docente: DAVID GONZALO CUSI QUISPE
Trimestre: Primer Trimestre

[Lista en PDF](#) ← Presionamos el Botón para descargar el archivo pdf

	(SABER/45pts)	(SABER/45pts)	(HACER/45pts)	(HACER/45pts)	(SER/DECIDIR/10pts)
 CUSI QUISPE DAVID GONZALO	2 Respuestas Calificación 32 pts	2 Aportes Calificación 36 pts	1 preguntas Calificación 45 pts	1 Tareas Calificación 32 pts	1 Notas Calificación 10 pts

Manual de Usuarios Administrativos

Proyecto: Aplicación Web para la Gestión de Procesos Educativos

Revisión 1

Autor: David Gonzalo Cusi Quispe



Aplicación Web Calama

Tabla de Contenido

1 Iniciar Sesión

2 Cargar Usuarios

3 Compartir Comunicados

El Alto, noviembre del 2020

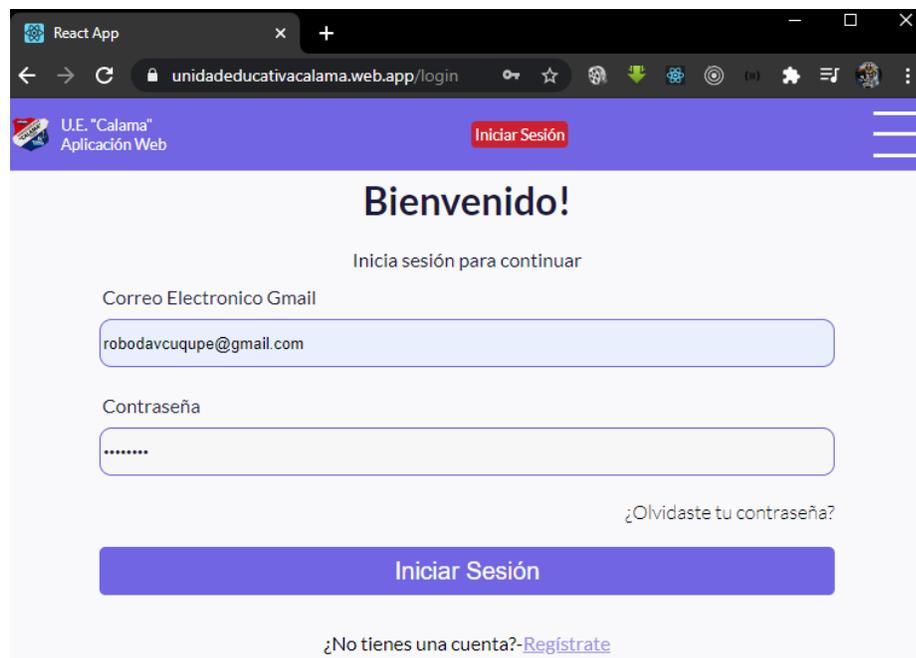
1 Iniciar Sesión

Para iniciar sesión en el sistema el usuario previamente debe tener una cuenta creada en el sistema.

Paso 1: Presionar el botón de Iniciar Sesión.



Paso 2: Una vez presionado el botón de Iniciar Sesión se mostrará un formulario de inicio de sesión en donde se debe ingresar su correo electrónico y la contraseña correspondiente.



Paso 3: Por ultimo debe presionar el botón de Iniciar Sesión.

React App

unidadeducativacalama.web.app/login

U.E. "Calama"
Aplicación Web

Iniciar Sesión

Bienvenido!

Inicia sesión para continuar

Correo Electronico Gmail

robodavcuque@gmail.com

Contraseña

.....

¿Olvidaste tu contraseña?

Iniciar Sesión

¿No tienes una cuenta?-[Regístrate](#) **Presionar el botón**

Una vez presionado el botón el sistema le mostrará el siguiente mensaje:

unidadeducativacalama.web.app dice

Bienvenido al sistema

Aceptar

2 Cargar Usuarios

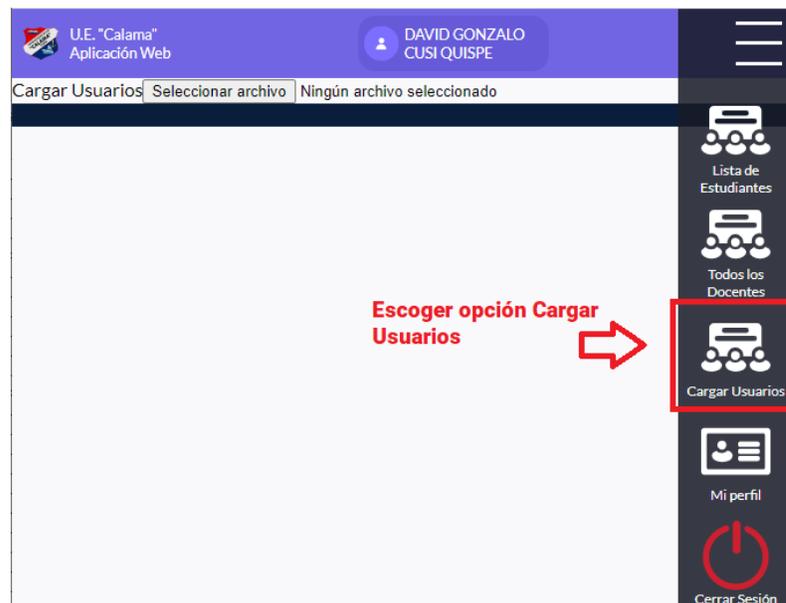
La operación de cargar usuarios permite al personal administrativo crear cuentas para todos los usuarios de la institución como ser: Docentes y Estudiantes. Para ello se debe de tener previamente un archivo en formato Excel.

En la siguiente figura se muestra el formato de las tablas y columnas del archivo Excel.

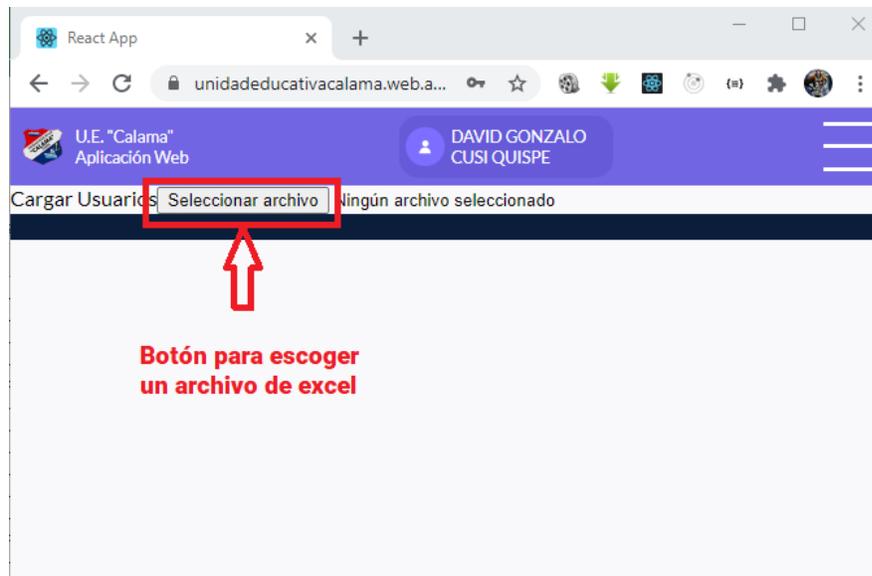
	A	B	C	D	E	F	G	H
1	Nro	apellidoPaterno	apellidoMaterno	nombres	email	password	curso	rol
2	1	ALIAGA	ALCON	JHUDIT BELEN	aliagaalconjhuditbelen@gmail.com	aliagaalconjhuditbelen	1 Primero A	estudiante
3	2	CALLE	HUANCA	ALEYDA	callehuancaaleyda@gmail.com	callehuancaaleyda	1 Primero A	estudiante
4	3	CHIPANA	TANTANI	ADRIAN LUDWIN	chipanatanianadrianludwin@gmail.com	chipanatanianadrianludwin	1 Primero A	estudiante
5	4	CHOQUE	SALINAS	GUILLERMO GONZALO	choquesalinasguillermogonzalo@gmail.com	choquesalinasguillermogonzalo	1 Primero A	estudiante
6	5	CHUQUIMIA	ALCON	EYRON MIJAE	chuquimiaalconeyronmijael@gmail.com	chuquimiaalconeyronmijael	1 Primero A	estudiante
7	6	FABIAN	CHUQUIMIA	IVAN ANDRES	fabianchuquimiaivanandres@gmail.com	fabianchuquimiaivanandres	1 Primero A	estudiante
8	7	FERNANDEZ	QUINTA	JENRY	fernandezquintajenny@gmail.com	fernandezquintajenny	1 Primero A	estudiante
9	8	FLORES	FUENTES	FLOR DE BELEN	floresfuentesflordebelen@gmail.com	floresfuentesflordebelen	1 Primero A	estudiante
10	9	GUACULLA	CUTIPA	ROSSI LUCERO	guacullacutiparossilucero@gmail.com	guacullacutiparossilucero	1 Primero A	estudiante
11	10	GUTIERREZ	APAZA	DENIS JHONNY	gutierrezapazadenisjhonny@gmail.com	gutierrezapazadenisjhonny	1 Primero A	estudiante
12	11	LAURA	FERNANDEZ	KATHERINE GABRIELA	laurafernandezkatherinegabriela@gmail.com	laurafernandezkatherinegabriela	1 Primero A	estudiante
13	12	MAMANI	ASISTIRI	MIJAE CRISTIAN	mamaniasistirimijaelcristian@gmail.com	mamaniasistirimijaelcristian	1 Primero A	estudiante
14	13	MAMANI	NINA	ROXANA	mamanininaroxana@gmail.com	mamanininaroxana	1 Primero A	estudiante
15	14	MENDO	MACHACA	DANIEL	mendomachacadaniel@gmail.com	mendomachacadaniel	1 Primero A	estudiante
16	15	MITA	VILLACORTA	RUDY	mitavillacortarudy@gmail.com	mitavillacortarudy	1 Primero A	estudiante
17	16	PAYE	LAUREL	MERY ARACELY	payelaurelmeryaracely@gmail.com	payelaurelmeryaracely	1 Primero A	estudiante
18	17	PLATTERS	LIMA	ADEDAIL ERNESTO	platterslimaadedailernesto@gmail.com	platterslimaadedailernesto	1 Primero A	estudiante
19	18	QUENTA	CONDORI	SINDEL ARIANA	quentacondorisindelariana@gmail.com	quentacondorisindelariana	1 Primero A	estudiante
20	19	QUISBERT	VALLEJOS	EMILY WARA	quisbertvallejozemilywara@gmail.com	quisbertvallejozemilywara	1 Primero A	estudiante
21	20	RAMOS	MAMANI	LUZMILA	ramosmamaniluzmila@gmail.com	ramosmamaniluzmila	1 Primero A	estudiante
22	21	SALGUEIRO	CALLISAYA	ADRIANA DANIELA	salgueirocallisayaadrianadaniela@gmail.com	salgueirocallisayaadrianadaniela	1 Primero A	estudiante
23	22	SIRPA	AJATA	MELVI MINEL	sirpaajatamelviminel@gmail.com	sirpaajatamelviminel	1 Primero A	estudiante
24	23	TRIGUERO	MAMANI	MARIA ELENA	trigueromamanimariaelena@gmail.com	trigueromamanimariaelena	1 Primero A	estudiante
25	24	TUMIRI	QUINTA	BRISA DAFNE	tumiriquintabrisadafne@gmail.com	tumiriquintabrisadafne	1 Primero A	estudiante
26	25	VILLAN	CONDORI	MARIA ANGEL	villancondorimarriaangel@gmail.com	villancondorimarriaangel	1 Primero A	estudiante
27	26	VILLAZANTE	MAMANI	DEVANI YANIRA	villazantemamanidevanianira@gmail.com	villazantemamanidevanianira	1 Primero A	estudiante

Los pasos para cargar usuario son los siguientes:

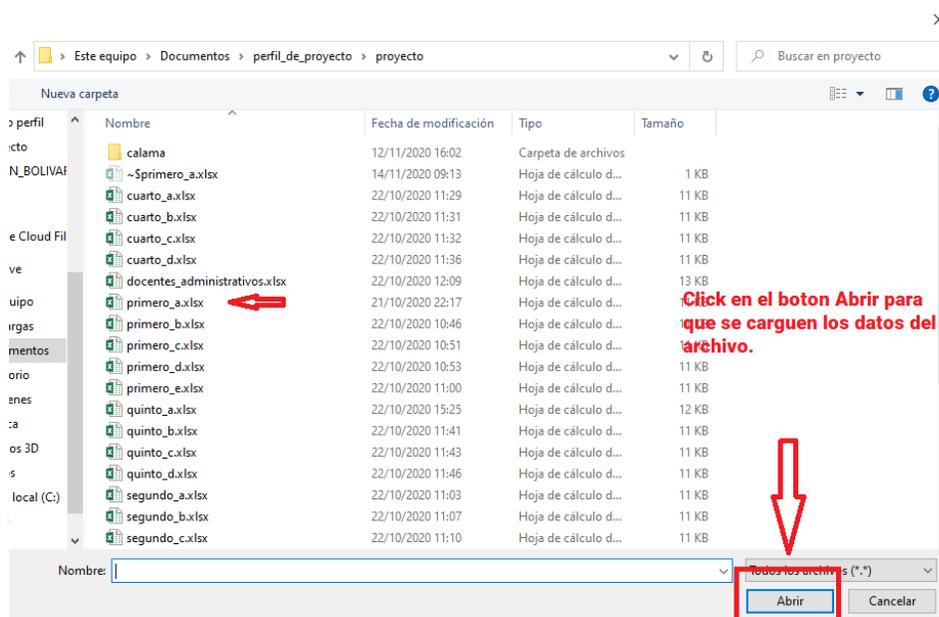
Paso 1: En las opciones del botón menú escoger **Cargar Usuarios**.



Paso 2: Seleccionar el botón de Seleccionar archivo.

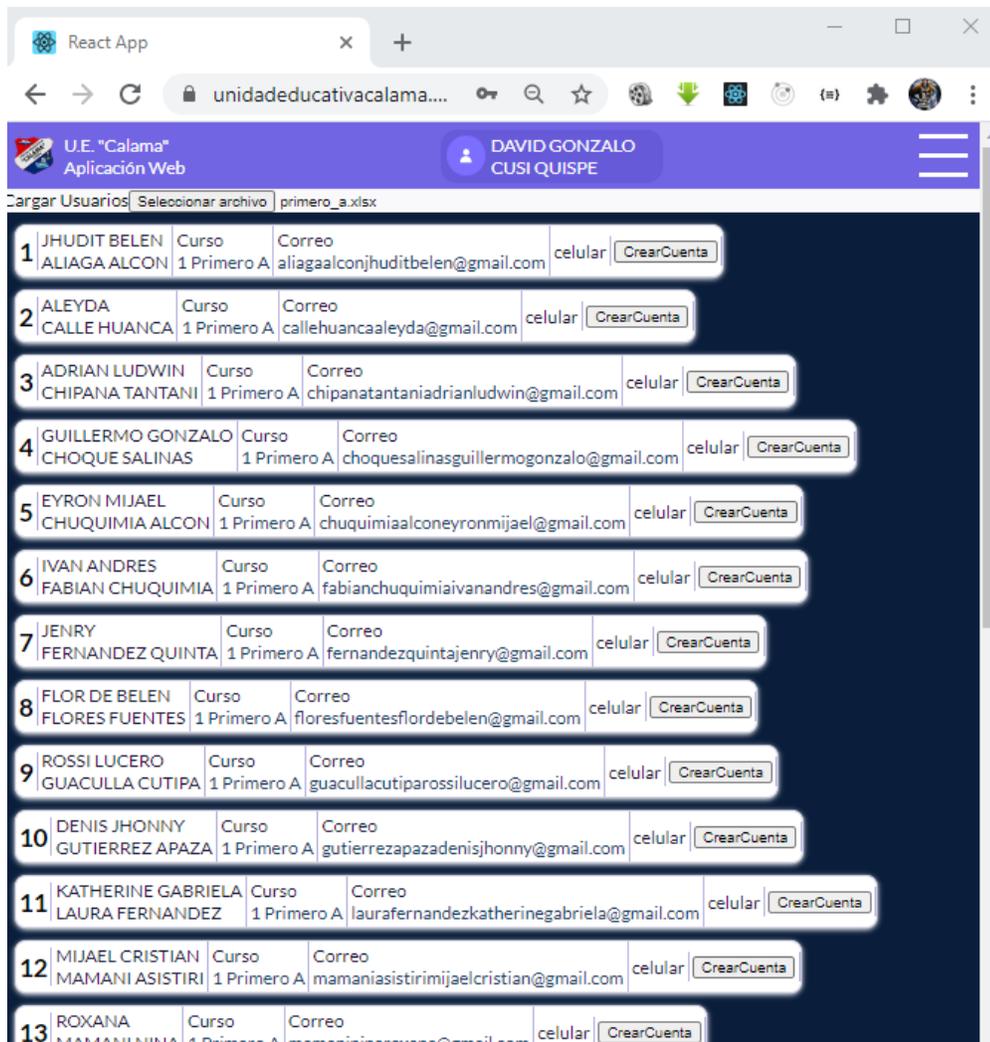


Paso 3: En la ventana emergente se selecciona el archivo en formato Excel.



Paso 4: Una vez seleccionado el archivo el sistema mostrara la misma lista de archivos con un botón de **Crear Cuenta** al final de cada fila.

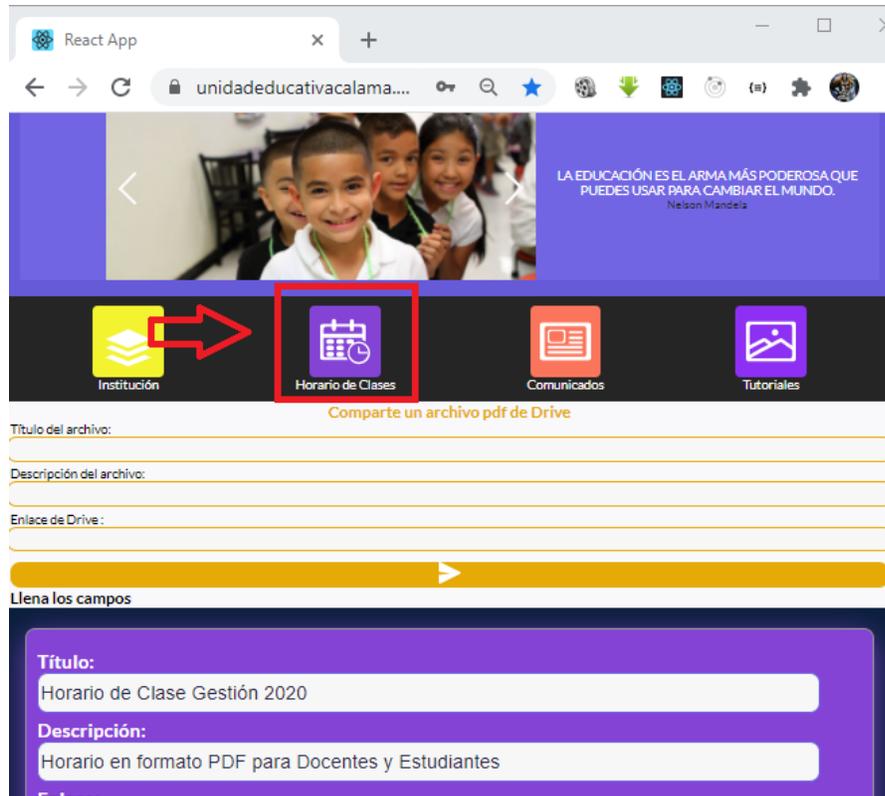
En esta parte el usuario crear las cuentas de los estudiantes que pertenecen a la unidad educativa.



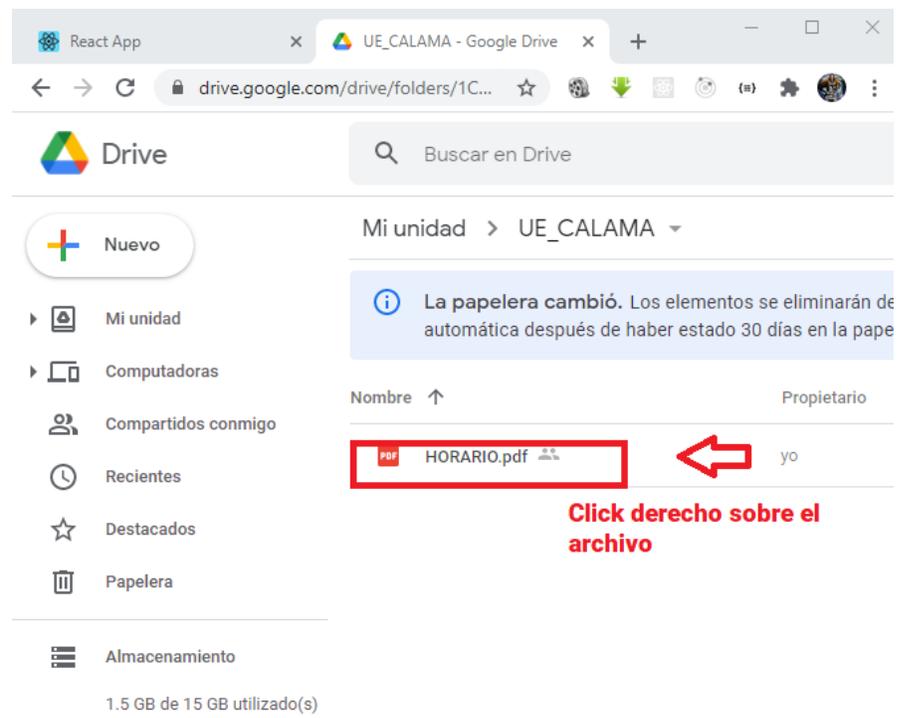
3 Compartir Horario de Clases

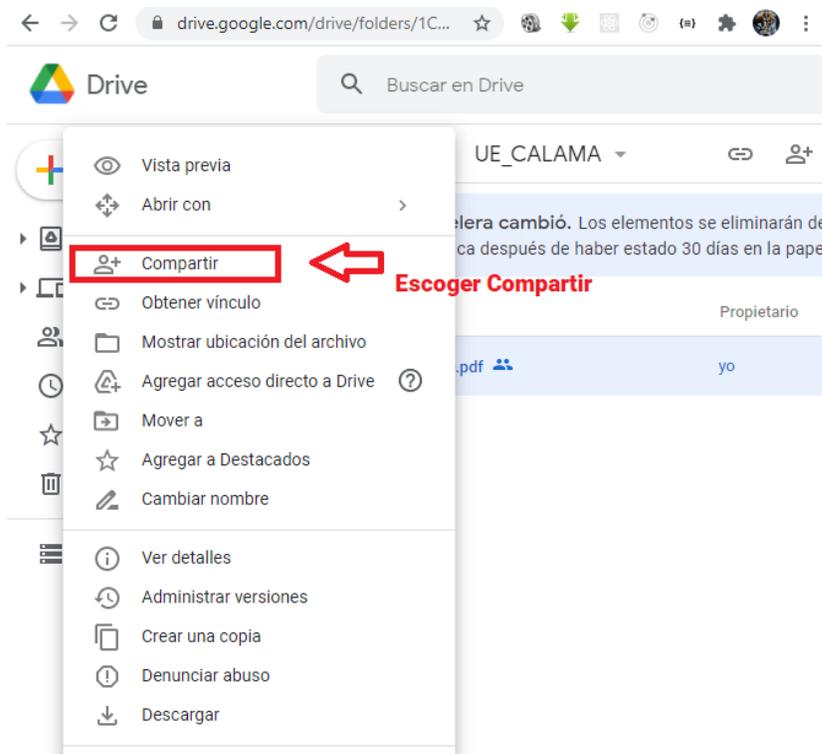
El sistema maneja en la sección de Horario de clases la opción de compartir un documento de Drive para los usuarios administrativos.

Paso 1: Seleccionar la opción de Horario de Clases en la pantalla principal.

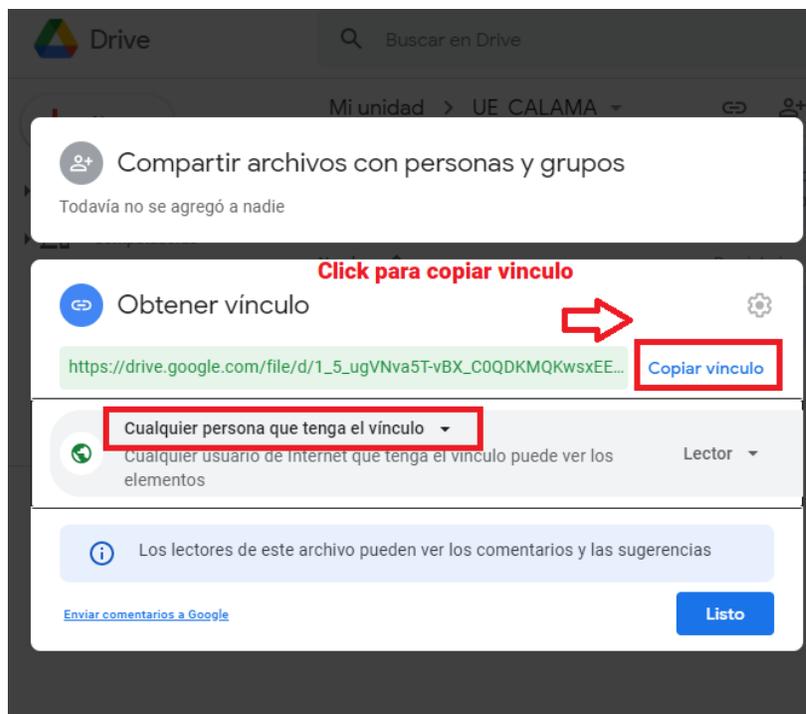


Paso 2: Copiar el enlace de un archivo de google drive.
En este ejemplo copiaremos el archivo HORARIO.pdf.

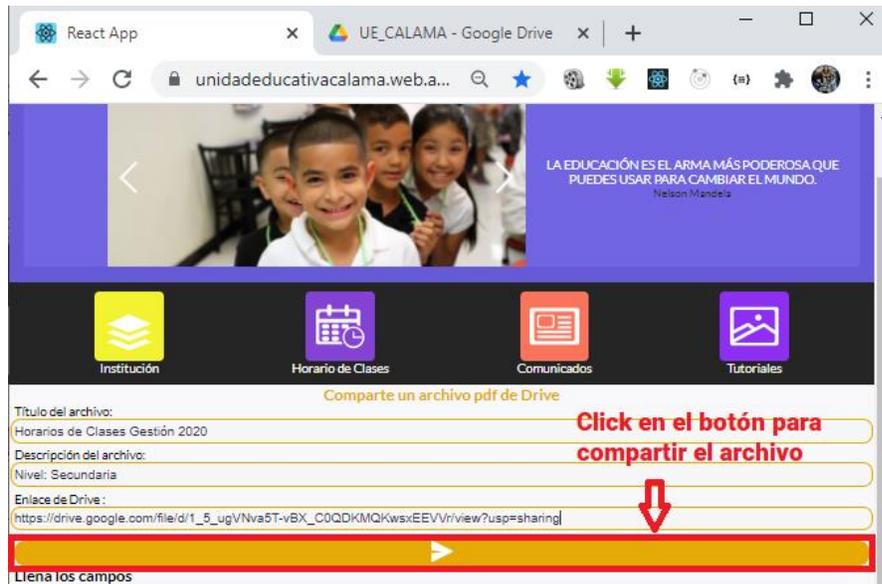




En este paso antes de copiar el vínculo debemos revisar que el este con la opción de Cualquier persona que tenga el vínculo como se puede observar en la imagen.

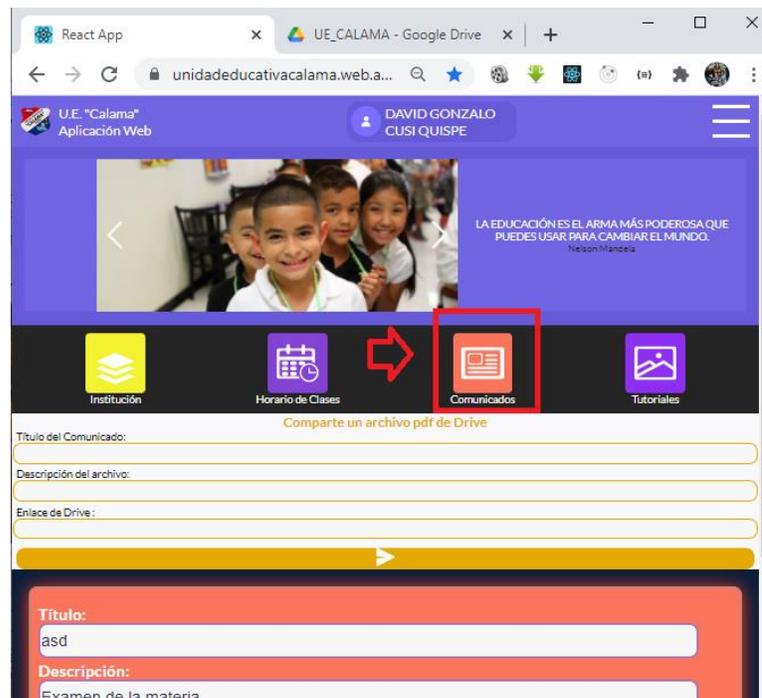


Paso 3: Llenar el formulario para Compartir un archivo.



3 Compartir Comunicados

Para compartir comunicados se tiene la sección de comunicados en donde encontraremos un formulario que debemos de llenar siguiendo los mismos pasos que se hacen para compartir Horarios de Clases.



ANEXO G: CUADRO DE EVALUACIÓN DEL DOCENTE

CUADRO DE EVALUACIÓN TRIMESTRAL - U. E. CALAMA - 2020												
Campo: CIENCIA TECNOLOGIA PRODUCCION			Año de Esc.: SEGUNDO A			Docente:						
Área: MATEMATICA			Nivel: Secundaria			Director/a:						
1er. TRIMESTRE			EVALUACIÓN - MAESTRA(O)									
Nro	Apellidos y Nombres	saber/45			Hacer/45			Ser-Decidir/10		Promedio	NOTA FINAL	
		Examen	Cuadernos	Promedio	Trabajos Pract	Ejercicios	Promedio	Asistencia	Participación			
1	BAUTISTA QUISEP ERIKA	92	92	41	92	94	42	95	95	10	93	
2	CATACORA VENTURA XIMENA	65	40	24	55	42	22	95	95	10	55	
3	CHAVEZ CHACHAHUAYNA FRANCYAMI	65	42	24	42	50	21	95	95	10	55	
4	COARITE VASQUEZ ROSALIA	65	50	26	60	60	27	100	100	10	63	
5	HUANCA ROJAS MAGALY	62	48	25	42	50	21	100	100	10	55	
6	LUPA LARICO NILDA ESTER										0	
7	MAMANI APAZA PAMELA	65	70	31	70	64	30	95	95	10	70	
8	POMA CHAVEZ DAFNE	100	100	45	100	100	45	100	100	10	100	
9	QUISEP APAZA EVA CRISTINA	65	50	26	65	56	27	95	95	10	63	
10	QUISEP APAZA WILLKAW QHANANCHI	100	100	45	100	100	45	100	100	10	100	
11	QUISEP HUALLPA LUZ NOEMI	65	75	32	75	75	34	80	80	8	73	
12	RAMIREZ FLORES KEVIN	92	92	41	92	92	41	95	95	10	92	
13	RAMOS VASQUEZ JHONATAN	100	100	45	100	100	45	100	100	10	100	
14	VALLEJOS QUISEP ADELA	65	79	32	85	85	38	95	95	10	80	
15	MAMANI HUANCA MARIA	65	50	26	65	50	26	95	95	10	62	
16												
17												
18												
19												
20												
21												

ANEXO H: Enlace a encuestas realizadas a docente y estudiantes de la Unidad Educativa Calama

Encuesta Hechas:

- https://docs.google.com/forms/d/12-RuE29tsSgxzY-9B4v3QRpb_Trh9ug1e7_6698y5tA/edit?usp=sharing
- https://docs.google.com/forms/d/1e1P42Sj2ahJBzeXq_NV1K_zkvWFH3Ik_oBUG4LRQP44/edit?usp=sharing

Resultados obtenidos:

- https://docs.google.com/spreadsheets/d/1iESsN_KHTdhCmYcOen-J7loKYfZTA5d9FEf-p3C7AYk/edit?usp=sharing
- https://docs.google.com/spreadsheets/d/19NMB1HGst0L5tmgvPp_iAlfgE75SLxcSHUo8cs8IOjU/edit?usp=sharing

ANEXO I: Aval de inicio de Proyecto

El Alto, 10 de septiembre del 2020

Señor:

Ing. David Carlos Mamani Quispe
DIRECTOR DE CARRERA INGENIERIA DE SISTEMAS

Presente. –

REF.: AVAL DE INICIO DE PROYECTO

Distinguido Director:

Mediante la presente tengo a bien comunicarle mi conformidad con el inicio del proyecto de grado **"APLICACIÓN WEB PARA LA GESTIÓN DE PROCESOS EDUCATIVOS"** que propone el estudiante **David Gonzalo Cusi Quispe** con cedula de identidad **7040904** expedido en la ciudad de La Paz para la Unidad Educativa "Calama" del turno Tarde.

Sin otro particular, reciba saludos cordiales.

Atentamente,




Lic. Wilfredo Villegas Alvarez
DIRECTOR DE LA U. E. "CALAMA"
C. I. LP

ANEXO J: Aval de Implementación de proyecto

El Alto, 18 de noviembre del 2020

Señor:

Ing. David Carlos Mamani Quispe
DIRECTOR DE CARRERA INGENIERIA DE SISTEMAS

Presente. –

REF.: AVAL DE LA IMPLEMENTACIÓN DEL PROYECTO DE GRADO “APLICACIÓN WEB PARA LA GESTIÓN DE PROCESOS EDUCATIVOS” PARA LA UNIDAD EDUCATIVA CALAMA TARDE

De mi mayor consideración

Por intermedio de la presente le hago llegar mis más cordiales saludos a su distinguida autoridad, esperando tenga éxitos en las funciones que desempeña en bien de nuestra sociedad.

Por la presente notificamos a usted que la Unidad Educativa Calama tarde extiende su **AVAL DE LA IMPLEMENTACIÓN DEL PROYECTO “APLICACIÓN WEB PARA LA GESTIÓN DE PROCESOS EDUCATIVOS” CASO: (UNIDAD EDUCATIVA “CALAMA”)** presentado por el Sr. David Gonzalo Cusi Quispe con C.I. 7040904 LP.

Teniendo a la vista el proyecto, declaramos tener los documentos referentes al proyecto de grado.

Sin otro particular, saludo a usted atentamente,



C.E. Wilfredo Villagás Alvarez
DIRECTOR
U.E. "CALAMA TARDE"

ANEXO K: Aval de Conformidad Tutor Metodológico

El Alto 25 de noviembre de 2020

Señores:

**HONORABLE CONSEJO DE CARRERA
CARRERA INGENIERÍA DE SISTEMAS**

Presente. –

Ref.: AVAL DE CONFORMIDAD

Distinguida autoridad

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado **"APLICACIÓN WEB PARA GESTIÓN DE PROCESOS EDUCATIVOS" CASO: UNIDAD EDUCATIVA "CALAMA"**, que propone el postulante David Gonzalo Cusi Quispe Vega con C.I.: 7040904 LP. Para su defensa pública, evaluación correspondiente a la materia "Taller de Grado II", de acuerdo a la reglamentación vigente de la carrera de Ingeniería de sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido con las consideraciones más distinguidas. Atentamente.



ING. MARICEL YARARI MAMANI

DOCENTE DE TALLER DE GRADO II

ANEXO L: Aval de Conformidad Tutor Revisor

El Alto, 25 de noviembre del 2020

Señor(a):

Ing. Maricel Yarari Mamani
**TUTOR METODOLÓGICO DE TALLER DE GRADO II
CARRERA DE INGENIERIA DE SISTEMAS - UPEA**

Presente. –

REF.: AVAL DE CONFORMIDAD DE PROYECTO DE GRADO

De mi mayor consideración:

Tengo a bien dirigirme a su persona, para darle a conocer que luego de efectuar el seguimiento a la estructura y contenido del Proyecto de Grado titulado: **“APLICACIÓN WEB PARA LA GESTIÓN DE PROCESOS EDUCATIVOS” CASO: UNIDAD EDUCATIVA “CALAMA”** elaborado por el Universitario David Gonzalo Cusi Quispe con C.I. 7040904 LP, en calidad de Tutor Revisor expreso mi conformidad con el contenido y la forma de trabajo, dando **mi Aval**, para que el postulante pueda realizar la defensa del Proyecto de Grado, para optar al título de Licenciado en Ingeniería de Sistemas, de acuerdo a normas y reglamentos vigentes.

Sin otro particular, me suscribo de su persona con las consideraciones más distinguidas.

ING. SERGIO RAMIRO ROJAS SAIRE
TUTOR REVISOR

ANEXO M: Aval de Conformidad Tutor Especialista

El Alto, 25 de noviembre del 2020

Señor(a):

Ing. Maricel Yarari Mamani
TUTOR METODOLÓGICO DE TALLER DE GRADO II
CARRERA DE INGENIERIA DE SISTEMAS - UPEA

Presente. –

REF.: AVAL DE CONFORMIDAD DE PROYECTO DE GRADO

De mi mayor consideración:

Tengo a bien dirigirme a su persona, para darle a conocer que luego de efectuar el seguimiento a la estructura y contenido del Proyecto de Grado titulado: **“APLICACIÓN WEB PARA LA GESTIÓN DE PROCESOS EDUCATIVOS” CASO: UNIDAD EDUCATIVA “CALAMA”** elaborado por el Universitario David Gonzalo Cusi Quispe con C.I. 7040904 LP, en calidad de Tutor Especialista expreso mi conformidad con el contenido y la forma de trabajo, dando **mi Aval**, para que el postulante pueda realizar la defensa del Proyecto de Grado, para optar al título de Licenciado en Ingeniería de Sistemas, de acuerdo a normas y reglamentos vigentes.

Sin otro particular, me suscribo de su persona con las consideraciones más distinguidas.



M. SC. REYNALDO JAMIER ZEBALLOS DAZA
TUTOR ESPECIALISTA